150 ptas.
mi computer
47
CURSO PRACTICO DEL ORDENADOR PERSON
EL MICRO Y EL MINIORDENADOR
PSION ORGANISER
Editorial Delta, S.A.

mi COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona

Volumen IV - Fascículo 47

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones, Francisco Martín
Asesor técnico: Ramón Cervelló

Redactores y colaboradores: G. Jefferson, R. Ford, F. Martín, S. Tarditti, A. Cuevas, F. Blasco
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, 08008 Barcelona
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador*, se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-7598-005-8 (tomo 4)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 058412
Impreso en España - Printed in Spain - Diciembre 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, 28034 Madrid.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Edificio Bloque Dearmas, final Avda. San Martín con final Avda. La Paz, Caracas 1010.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)
Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 6.850.277 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Editorial Delta, S.A. (Paseo de Gracia, 88, 5.°, 08008 Barcelona), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Editorial Delta, S.A., en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.





Compresor de datos

"Organiser" (organizador) es un revolucionario y eficaz microordenador que puede caber en el bolsillo del usuario

Ian McKinnell

El Psion Organiser tiene el mismo tamaño y peso que una máquina de afeitar eléctrica y a primera vista tiene precisamente ese aspecto. Una carcasa rígida envuelve el teclado tipo calculadora, que posee 36 pequeños botones. Éstos incluyen las letras del alfabeto (el Organiser sólo utiliza caracteres en mayúsculas) y varias teclas de función. A los dígitos del 0 al 9, a algunas funciones matemáticas y a los signos de puntuación se accede pulsando la tecla Shift (de cambio). Encima del teclado hay una visualización en cristal líquido de 16 caracteres, que dispone de un contraste regulable de modo que se la puede leer desde cualquier ángulo. Cada carácter se compone de una matriz de puntos de cinco por ocho.

En la parte posterior de la máquina hay dos ranuras para insertar los "paquetes de datos" que proporcionan el almacenamiento a largo plazo. El Organiser viene equipado con paquetes de datos de ocho Kbytes, aunque también hay a la venta unos paquetes de 16 Kbytes.

Cuando se pulsa el botón ON de la parte superior izquierda del teclado, la visualización muestra la hora en un reloj de 24 horas y la fecha. Las cuatro funciones principales del Organiser se obtienen mediante la pulsación de un único botón. Como explica claramente el manual, no hay necesidad de teclear instrucciones. Estas funciones principales son:

- SAVE: Desde el teclado se pueden introducir registros de datos de hasta 200 caracteres de longitud, editarlos con los dos controles del cursor (izquierda y derecha) y la tecla Delete (borrar), y después guardarlos (SAVE) en cualquiera de los paquetes de datos.

Archivo rápido
En Psion Organiser, concebido como un ordenador de bolsillo con una poderosa base de datos, proporciona un sistema de archivo rápido e instantáneo. Con funciones de calculadora incorporadas y la capacidad de almacenar y recuperar información vital con suma rapidez, el Organiser puede manipular algunas de las funciones normalmente restringidas a microordenadores de escritorio o manuales más grandes

• FIND: Esta función recupera datos desde cualquiera de los paquetes de datos. Se puede efectuar la búsqueda de una serie de hasta 15 caracteres. La serie requerida se entra sobre la visualización y la máquina la compara con los registros almacenados. La instrucción FIND es muy rápida: el tiempo típico para buscar en una base de datos de 21 Kbytes es de ocho segundos. Los registros que contienen la serie requerida se visualizan por el orden en que están dispuestos. Usando las teclas del cursor se puede hacer rotar un registro en cualquier sentido.

Hemos utilizado el paquete de datos de Psion llamado *Restaurant guide to London* (Guía de restaurantes de Londres), una base de datos de 16 Kbytes que contiene los nombres de 105 restaurantes, cada uno de ellos con dirección, número de teléfono, estación de metro más cercana, tipo de comida, horario de atención al público, orientación sobre precio aproximado del cubierto y un comentario general. Una búsqueda de la serie "ARB" da como resultado un restaurante llamado "BARBARELLAS" y también un restaurante próximo a la estación de metro de "MARBLE ARCH". Esta función se puede comparar favorablemente, en cuanto a velocidad y conveniencia, con muchos programas de base de datos para microordenadores.

• ERASE: Elimina un registro determinado de su paquete de datos. En el teclado del Organiser no hay ninguna tecla de función ERASE (borrar) y a esta instrucción sólo se puede acceder después de haber utilizado la instrucción FIND. Con el registro en la visualización, el usuario pulsa la tecla Mode (modalidad) para obtener la facilidad ERASE. Pulsando entonces el botón Execute (ejecutar), el registro se borrará de la memoria. Sin embargo, esto no deja libre ningún espacio de almacenamiento para su reutilización, sino que tan sólo hace que el archivo quede inaccesible para la CPU. Más adelante en este mismo capítulo analizaremos la naturaleza de la memoria de los paquetes de datos.

• CALC: Mediante esta instrucción se puede utilizar el Organiser como una calculadora de cuatro funciones (suma, resta, multiplicación, división). El resultado da siete cifras significativas y se utiliza la

Sistemas contrastados

A fines de evaluación, hemos comparado al Psion Organiser con un sofisticado sistema de agenda/diario/notas de hojas sueltas

	PSION	FILOFAX
DIMENSIONES	142×78×29,3 mm	180×124×30 mm
PESO	225 g	200 g
VISUALIZACIÓN	LCD 16 caracteres	Entrada manuscrita-mecanografiada por el usuario
ALMACÉN.	10 900 caracteres por rampak de 8 Kbytes	Hasta 200 páginas
FACILIDADES DE BÚSQUEDA	A partir de cualquier pista de hasta 15 caracteres	Manual, índice alfabético
PAQUETES DISPONIBLES	Rampak de 8 o 16 Kbytes; científicos, ingeniería, matemáticas, contabilidad y financieros, relación de restaurantes	Diario, páginas de teléfonos-direcciones, contabilidad de costos, hoja electrónica, mapas, papel para apuntes y papel pautado para música
OTRAS FACILIDADES	Dice la hora y la fecha. Se puede utilizar como calculadora	Viene en una elegante cartera de piel con lugar para almacenar tarjetas comerciales y de crédito



Potencia a mano
El teclado Psion contiene el abecedario completo, dispuesto por orden alfabético. Este trazado no es intuitivo y puede plantear dificultades a quienes estén habituados a un teclado QWERTY. La tecla de cambio (SHIFT) proporciona acceso a un teclado numérico completo, funciones de calculadora, instrucciones para movimiento del cursor y para la base de datos incorporada

Kevin Jones

notación científica para números muy grandes o muy pequeños. Nuevamente, el teclado del Organiser no dispone de ninguna tecla etiquetada como CALC: a ella se accede pulsando la tecla Mode en cualquier momento, excepto después de haber ejecutado una búsqueda FIND.

La simplicidad y la sencillez de uso de la máquina corren parejas con su reducido precio. El Organiser es viable comercialmente porque la tecnología de baja potencia también es de bajo costo. La máquina puede funcionar mediante una sola pila PP3 de nueve voltios hasta seis meses, porque el sistema de circuitos utiliza chips CMOS (*Complementary Metal Oxide Semiconductor*: semiconductor de óxido metálico complementario) y los paquetes de datos son chips EPROM (*Erasable Programmable Read-Only Memory*: memoria programable de lectura solamente, que puede borrarse), ambos con niveles muy reducidos de consumo de potencia. Hasta hace muy poco, el costo de estos tipos de chip habría significado un precio demasiado alto, que hubiera comprometido su buena comercialización.

En el corazón del Organiser hay un procesador de ocho bits Hitachi 6301X que funciona a 0,93 MHz y está controlado por cuatro Kbytes de ROM. También hay dos Kbytes de RAM para los datos introducidos, información en pantalla y "espacio de trabajo" de la calculadora. Los paquetes de datos EPROM que se introducen en la parte inferior de la carcasa son el equivalente de la memoria RAM de otros microordenadores. Se puede pensar en la EPROM como una RAM "de una sola vez": los datos en ella escritos se almacenan y no se pueden quitar, si bien se puede acceder a ellos como si fuera una ROM normal. Por lo tanto, es una ROM "programable". Utilizando la función ERASE del Organiser en un registro del chip no se libera su espacio de almacenamiento en la EPROM, sino que se marca el registro para indicarle al procesador que está eliminado. Por consiguiente, los paquetes de datos se van llenando hasta que finalmente no se pueden agregar registros nuevos. La única forma de borrar los datos almacenados y restaurar el chip al estado "limpio" original es exponerlo a una luz ultravioleta intensa. Ésta es la función del Psion Formatter. Los chips de paquetes de datos nuevos se pueden formatear hasta 100 veces.

La ventaja que tiene la EPROM sobre la RAM es que, una vez que los datos han sido almacenados, no es necesaria una fuente continua de potencia para mantenerlos, mientras que la RAM es "volátil" y requiere un consumo constante de energía eléctrica. Asimismo, los chips de EPROM son muy fiables, y el costo de cada byte representa una quinta parte del de la RAM. Los utilizan comúnmente los fabricantes de ordenadores en los prototipos de sistemas en vez de ROM, porque en el caso de que se produzcan errores de software, los chips se pueden volver a utilizar y a programar, mientras que una ROM programada erróneamente es irrecuperable. La ROM se puede emplear después de que se hayan descubierto todos los fallos en el software contenido en EPROM.

La comunicación con otros dispositivos informáticos se consigue mediante una interface RS232; ésta se coloca en una de las ranuras para paquetes de datos y permite transmitir o recibir información a velocidades de hasta 9 600 baudios. Los protocolos se pueden programar utilizando el software que se suministra con la interface, de modo que, por ejemplo, se puede formatear la salida a una impresora por longitud de páginas y anchura de línea.

El Organiser se puede emplear en una gran variedad de situaciones para las que se necesita un ordenador potente y portátil. Se podría utilizar para anotar resultados experimentales o para llevar una agenda de citas, aunque el teclado limita seriamente la velocidad de digitación porque está dispuesto de forma alfabética. La mejor forma de utilizar la máquina es la de almacenar información que requiera una gran cantidad de referencias cruzadas (p. ej., listas de precios de vendedores, la relación indexada de una biblioteca, o los nombres, direcciones y números de teléfono de las amistades). También puede ser útil para aplicarle un programa de ordenador a datos experimentales sobre la marcha, o para almacenar información con el fin de procesarla luego en otro lugar.

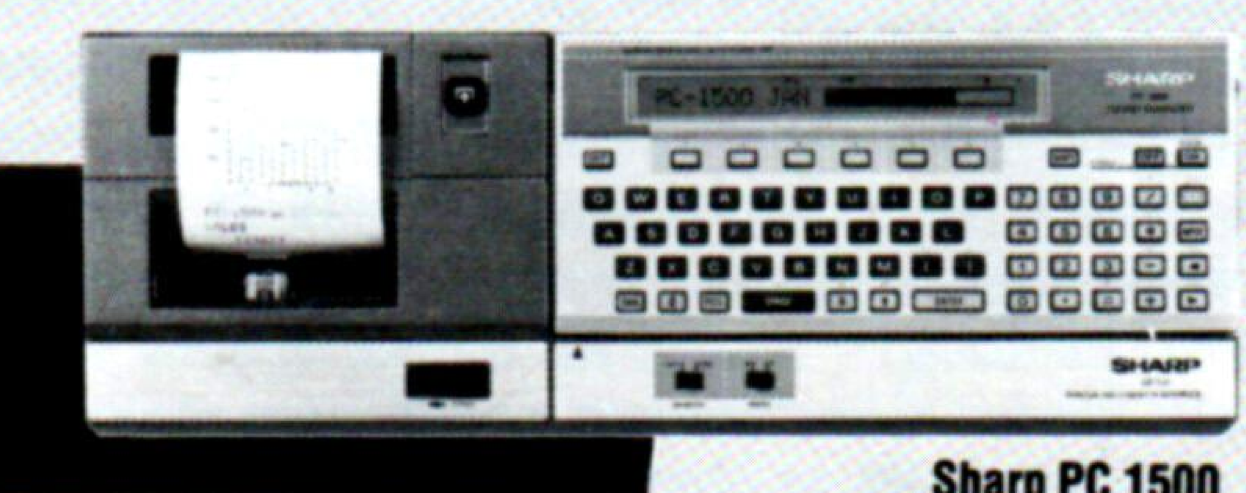
Ordenadores de bolsillo

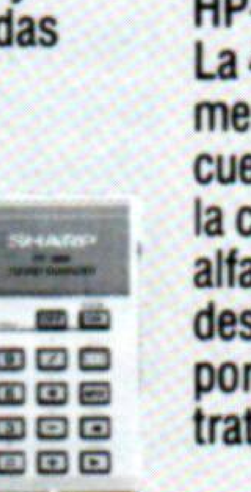
Durante estos últimos años se han realizado intentos por comercializar ordenadores de bolsillo de una u otra clase. Éstos se han vendido moderadamente bien, pero nunca han logrado ni siquiera una fracción de las ventas de la calculadora de bolsillo. Algunos ordenadores de bolsillo son poco más que calculadoras con pretensiones. Éstos se pueden programar en sus propios lenguajes, pero suelen atraer sólo a ingenieros y científicos que necesitan un medio de realizar complicados cálculos sobre la marcha. Por encima de ellos, el siguiente peldaño es el de los ordenadores de bolsillo que se pueden programar en BASIC. Este hecho los hace más adecuados para fines generales y, por consiguiente, se han realizado para ellos programas para diversos usos.

Ni siquiera los mejores ordenadores de bolsillo se han hecho auténticamente populares. No se ha escrito software para permitir que la gente lleve con ellos sus diarios de citas ni sus agendas de direcciones. Y aunque así se hubiera hecho, las memorias de los ordenadores suelen ser demasiado pequeñas como para retener una cantidad útil de información. Los teclados, a causa de su reducido tamaño, son difíciles de utilizar, de modo que la mayoría de las personas prefieren usar lápiz y papel para sus diarios y agendas. Las versiones informatizadas de estos dos objetos sólo ofrecen una ventaja cuando son precisas sofisticadas facilidades de búsqueda



Hewlett Packard 41C
Ésta es una calculadora programable muy sofisticada. Está a la venta en tres modelos: HP-41C, HP-41CV y HP-41CX. La 41CX tiene 0,5 K de memoria; los otros dos modelos cuentan con 2 K. A pesar de que la calculadora posee teclado alfabético, éste sólo está destinado a la programación y, por lo tanto, la calculadora no trata textos





Sharp PC 1500

Sharp PC 1251
Sharp fabrica dos ordenadores de bolsillo, el PC 1251 y el PC 1500. El primero es el ordenador de bolsillo más pequeño que existe en el mercado, y el segundo es el más potente. Ambos se programan en BASIC, si bien el PC 1500 posee instrucciones extras. Los dos tienen 4 K de memoria estándar, pero el PC 1500 se puede ampliar a 20 K



Texas Instruments TI-66
Se trata esencialmente de una calculadora programable y, como tal, es atractiva para científicos y matemáticos. No trata textos, de modo que no es adecuada para aplicaciones de carácter más general. Sólo posee 0,5 K de memoria



Casio FX700P
Este es uno de los varios ordenadores de bolsillo fabricados por Casio que se pueden programar en BASIC. Tiene 2 K de memoria y un diminuto teclado alfabético. También se fabrica una versión con una pequeña impresora térmica incorporada

Manos ganadoras

El bridge es un juego de naipes que despierta en los aficionados un apasionado interés y se presta muy bien a ser informatizado

Los programas de bridge disponibles a nivel comercial pertenecen a dos categorías claramente definidas, según estén diseñados como medios de enseñanza o como paquetes para jugar.

Es posible hallar en el mercado algunos buenos paquetes de enseñanza. Uno de los mejores, desde el punto de vista del diseño y la presentación, es la serie Bridgemaster. Existen versiones de este paquete para el Spectrum, ZX81, BBC Modelo B, Electron y Commodore 64. El diseño de esta serie pertenece a Terence Reese, ex campeón del mundo y autor de numerosos libros sobre este juego de naipes.

El principio en el que se basan todos los programas de enseñanza de bridge es que el principiante vaya pasando por una serie de manos preestablecidas escritas en el programa. Ésta es la diferencia esencial entre un paquete de enseñanza y un paquete para jugar, que genera manos al azar. Dado que el tutor "sabe" lo que contiene cada una de las manos, el programa es capaz de guiar al jugador paso a paso a través de las muchas reglas y convenciones que hacen que el bridge sea un juego tan atractivo y exigente desde el punto de vista intelectual.

La serie Bridgemaster proporciona un excelente ejemplo de lo que se puede conseguir gracias a un paquete de enseñanza. El cursor consta de dos programas, dirigidos, respectivamente, al principiante absoluto y al jugador aficionado. El primero se denomina *Complete learning package for the beginner at bridge* (Paquete completo de aprendizaje para la iniciación en el bridge); el segundo es el *Expert bridge* (Bridge para expertos). El paquete para principiantes se compone de dos cassettes de programas, dos cintas de comentarios, un escueto folleto de instrucciones y *Begin bridge*, libro de bolsillo escrito por Reese. Este último no se utiliza al mismo tiempo que los programas en la pantalla. Las cintas de comentarios proporcionan todo el apoyo que necesita el principiante.

Los programas de enseñanza de bridge que no poseen comentarios basados en cintas han de proporcionar un manual muy exhaustivo o bien observaciones sumamente explicativas en la pantalla. Por otra parte, también podrían presuponer al menos un conocimiento básico de bridge por parte del usuario y limitarse a mejorar el nivel de juego del mismo.

Para quienes no saben jugar al bridge, la baraja se reparte entre los cuatro jugadores que se agrupan en equipos de dos. En el bridge por ordenador, se adopta generalmente la convención estándar de designar a los cuatro jugadores (Norte, Sur, Este y Oeste). La postura precede a la partida, y en ella los jugadores, sin mostrarse el juego entre sí, deben declarar la fuerza de los naipes que tienen de mano. La fuerza de una mano está determinada por la cantidad de cartas altas o la cantidad de cartas de un palo o ambas. En nuestro diagrama, mostramos un ejemplo de postura. La postura ganadora determinará el contrato. Éste selecciona qué color ha de ser el triunfo así como la cantidad de bazas que el bando ganador debe intentar hacer. Después de la postura, se juega la mano. En la partida, el jugador que ha ganado la postura se convierte en el "declarante" y su pareja en el "muerto". Los naipes del "muerto" se ponen a la vista de todos. El declarante ganará o perderá puntos según se haga o no con la cantidad de bazas especificadas en el contrato.

Todos los programas de enseñanza de bridge representan la postura, tanto si permiten que el jugador la gane como si no. En Bridgemaster, al jugador se le da la oportunidad de hacer su propia postura, habiendo visto visualizados sus naipes en la pantalla. El ordenador declara entonces la postura que se ha de realizar. Ésta podría diferir de la postura escogida por el jugador. El programa de Reese, al igual que muchos programas de enseñanza, está arreglado de modo que el ordenador sólo acepte la postura o la partida que el diseñador del programa desea que uno haga. Reese admite que, en ocasiones, esto lleva a que se rechace una jugada perfectamente sensata del jugador en favor de la ruta predeterminada que se ha trazado de antemano.

Además de visualizar su mano y la postura, el programa debe manejar la jugada de la mano. El método de visualización más común es un cuadrado

Postura ganadora
Los valores de los naipes de la postura son: As-4; Rey-3; Reina-2; Valet-1. Las letras en rojo corresponden a la primera postura de los jugadores y las azules representan la segunda

en el centro de la pantalla con los cuatro lados señalados N, E, S y O. Luego se visualizan al completo las manos del Norte y el Sur. Por lo general se muestran los cuatro palos: piques, corazones, diamantes y tréboles, dispuestos por ese orden, con los valores de los naipes visualizados a los lados.

Se supone que el jugador es Sur y que el Norte o bien el Sur serán el «muerto». En cualquier caso, Sur siempre juega los naipes de ambos. Dicho en otras palabras, Sur juega siempre la mano, con independencia de que la postura la haya ganado Norte o Sur. Esto, por supuesto, constituye un alejamiento del verdadero bridge, en el cual un jugador se puede pasar toda la tarde sin ganar el contrato ni siquiera una sola vez.

En un programa de enseñanza esta diferencia no tiene relevancia, excepto en la importante área del juego defensivo. Se necesitan estas tácticas cuando uno desea impedir que sus oponentes cumplan el contrato después de haber ganado la postura. En un programa para jugar al bridge, las consecuencias son más graves y nosotros no conocemos ningún programa para jugar al bridge que juegue una secuencia en la que Este y Oeste ganen la postura. Si se evita la postura en, por ejemplo, el excelente programa de bridge de CP Software para el Spectrum de 48 Kbytes (es decir, si se permite de forma deliberada que Este u Oeste sean el declarante), el programa imprimirá en la pantalla un mensaje indicándole que no está diseñado para jugar en esas circunstancias e indicándole que pulse la tecla R para generar una mano nueva.

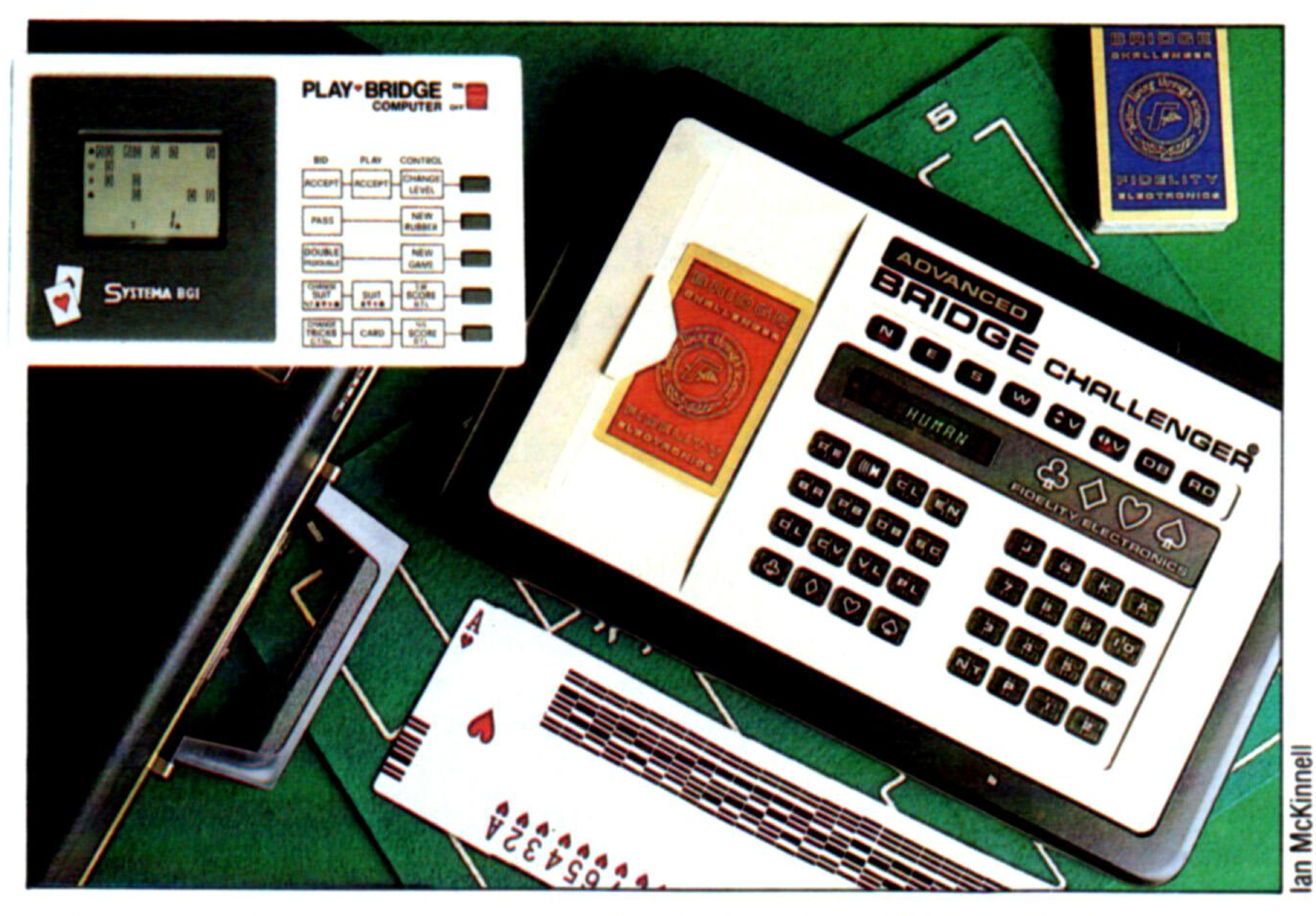
Cuando se ha completado la postura, se visualiza el juego de la mano. De acuerdo a quién gane la baza, el ordenador (que juega por E y O) o bien el jugador (que juega por N o S) selecciona un naipe. Los naipes de E y O aparecen en la pantalla de a uno, tal como sucedería en la mesa en el juego normal.

Los programas basados en ordenador ofrecen una gran cantidad de características útiles. Bridgemaster, por ejemplo, permite que el jugador elija entre cuatro opciones antes de cada turno de dar, según el nivel de competición. P juega las manos con el jugador humano seleccionando juegos para N y S (guiado por Reese); A selecciona *autoplay* (automático), en la que la mano termina sin su colaboración; H visualiza en la pantalla las cuatro manos, mientras que D selecciona un turno diferente. Este juego de opciones es el habitual en la mayoría de los programas de bridge.

La facilidad *autoplay* es especialmente útil en combinación con el comentario de Reese, porque el jugador no se tiene que distraer con la tarea de seleccionar naipes. Todos los turnos se pueden volver a jugar una y otra vez. Los programas para jugar al bridge también disponen de estas características, pero en este caso no actúan como repetición de ejercicio en ciertos puntos del aprendizaje, sino como *post mortems* de la postura y la partida.

Hemos analizado algunos de los principios de los paquetes de enseñanza, pero existen también en el mercado algunos programas para el jugador más avanzado que desea perfeccionar su juego.

El paquete *Expert bridge* de la serie Bridgemaster que hemos comentado previamente es un curso de instrucción avanzado bien presentado que analiza las complejidades de incidencias como manos de alta puntuación, "slam", aprietos y bloqueos.



Ian McKinnell

Desafío exclusivo
Las dos máquinas de la fotografía son ordenadores para jugar al bridge. El primero (derecha) es un juego de mesa llamado Advanced Bridge Challenger, de Fidelity Electronics. Bridge Challenger está recomendado sólo para jugadores de bridge experimentados, porque su dominio exige mucha práctica y concentración. Un gran inconveniente del Bridge Challenger es su diminuta pantalla de ocho caracteres. La segunda máquina (izquierda) es el ordenador Systema Play Bridge. Funciona a pilas. Comprender el Play-Bridge sólo ocupa algunos minutos; no posee ninguna característica avanzada, pero jugar con él puede ser apasionante

Otro paquete que merece la atención del jugador más experimentado es el programa para jugar al bridge de Alligata Software. Existen versiones para el Oric, BBC Modelo B, Electron y Commodore 64. El programa presupone cierto conocimiento de bridge por parte del jugador y con este paquete no se suministra ningún manual, de modo que los jugadores que no tengan buena memoria deberán disponerse a tomar notas.

El programa de CP Software para el Spectrum es muy entretenido para jugar. No se juega con la mano como declarante, de modo que no se puede practicar un juego defensivo. No obstante, como la mayoría de los jugadores aficionados, que constituyen el 98 % de la comunidad de jugadores de bridge de todo el mundo, desean jugar todas las manos de cualquier manera, éste no se considera un inconveniente serio.

Los programas de bridge, tanto los de aprendizaje como los de juego, constituyen una manera ideal de aprender a jugar y de mejorar los puntos débiles que uno tenga. El bridge es un juego que exige muchas habilidades diferentes: una buena memoria, aptitudes analíticas, capacidad para trabajar en pareja, perspicacia para jugar con cautela y saber cuándo asumir un riesgo, y habilidad para mantenerse inescrutable y conservar la cabeza fría. Los juegos que hemos analizado en este capítulo no simularán a la perfección una partida de bridge entre cuatro jugadores; pero, por cierto, le proporcionarán una práctica valiosa y estimulante.

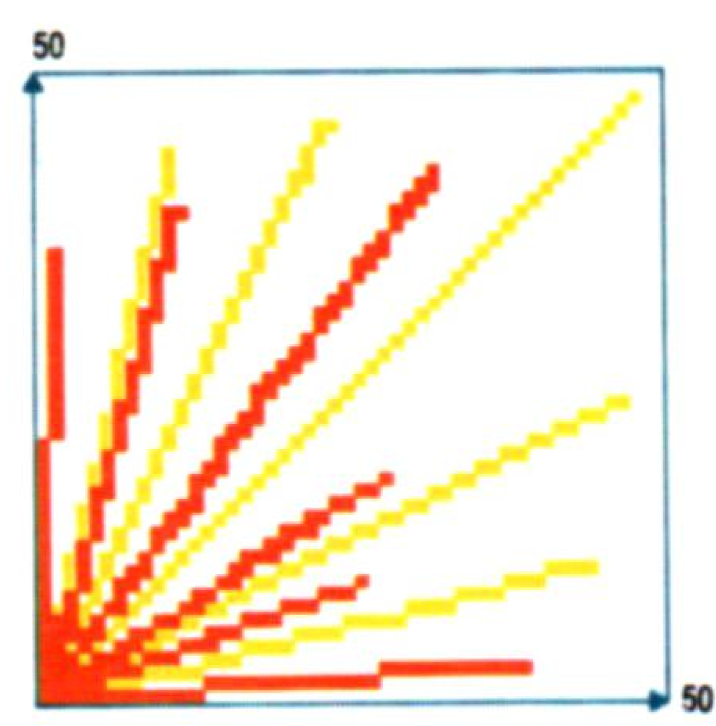


Ian McKinnell

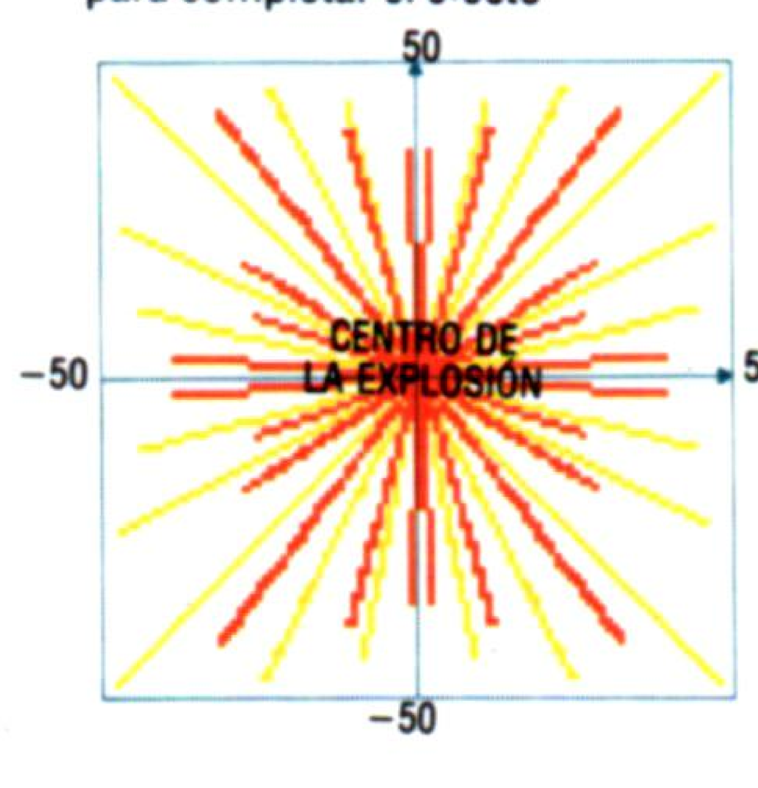
Elija un paquete
Existen varios paquetes de bridge a la venta, que van desde programas pedagógicos a complicados juegos de desafío que ponen a prueba la pericia de los jugadores más experimentados.
En la fotografía vemos (de izquierda a derecha) *Bridge master*, disponible en cassette para el Spectrum, ZX81, BBC Modelo B, Electron y Commodore 64; *Bridge*, de Alligata Software, en cassettes para el Oric, BBC Modelo B, Electron y Commodore 64; y *Bridge player*, de CP Software, que se vende en cassette para el Spectrum de 48 Kbytes

Explosión

He aquí los procedimientos destinados a crear los efectos visuales y sonoros de una explosión



Jugada explosiva
Los gráficos se crean generando líneas de longitud al azar comprendidas entre una escala de −50 a 50 unidades desde el centro. La explosión se construye por cuadrantes y se reproduce a modo de espejo para completar el efecto



En el último capítulo de nuestro proyecto para crear el juego "Campo de minas", destinado al BBC Micro y al Electron, estudiamos cómo el programa detectaba las colisiones con las minas. En esta ocasión analizaremos los procedimientos que nos permitirán crear los efectos visuales y sonoros de las explosiones.

El BBC Micro y el Electron admiten 16 posibles variaciones de color que podemos utilizar en la creación de un efecto de explosión. En realidad sólo hay ocho colores distintos, siendo las otras ocho variaciones efectos de intermitencia (*flashing*) entre un color y otro. Normalmente, las combinaciones intermitentes de colores se alternan cada medio segundo, pero se puede cambiar esta periodicidad mediante dos instrucciones FX. *FX9 establece la duración del tiempo durante el cual se visualiza el primer color del par intermitente. El tiempo se mide en unidades de cincuentavos de segundo. Por consiguiente, *FX9,20 altera el tiempo de visualización del primer color a 2/5 (dos quintos) de segundo. La otra instrucción FX, *FX10, se puede utilizar para alterar el tiempo de visualización del segundo color de la misma manera. Modificando uno o ambos tiempos de visualización se pueden producir interesantes efectos centelleantes.

El efecto visual de la explosión se puede producir dibujando una serie de líneas cortas en diversos colores que partan desde un punto central. Para mejorar el efecto, también se puede cambiar rápidamente el color de las minas y de las letras del marcador. Como éstos se imprimirán originalmente en el color lógico 2 (que hemos establecido en verde) podemos utilizar la siguiente instrucción para reasignarle al color lógico 2 un color seleccionado al azar entre las 16 posibles combinaciones:

```
VDU19,2,RND(15),0,0,0
```

RND(15) selecciona un número entero entre 1 y 15. Ello significa que el color 0 no se selecciona nunca; pero como el color 0 es negro, el color del fondo, esto no tiene importancia alguna.

El color utilizado para trazar las líneas de la explosión también se puede seleccionar al azar utilizando GCOL 0,RND(3). Luego empleamos un bucle para dibujar una serie de líneas desde el centro de la explosión en una serie de colores seleccionados al azar. Hasta ahora hemos analizado las instrucciones de alta resolución MOVE y DRAW. DRAW(X,Y) siempre traza una línea hasta el punto con coordenadas "absolutas" (X,Y). Sin embargo, existe otra familia de instrucciones para dibujar en alta resolución que nos permiten, entre otras cosas, especificar puntos "relativos" entre sí. Se puede emplear la instrucción PLOTK,X,Y para dibujar líneas entre puntos relativos o absolutos en función del valor de K. La siguiente tabla muestra algunas de las variaciones que son posibles utilizando PLOT:

K=0	Movimiento relativo al último punto
K=1	Trazar línea hasta posición relativa en color de primer plano
K=2	Trazar línea hasta posición relativa en color lógico inverso
K=3	Trazar línea hasta posición relativa en color de fondo
K=4	Movimiento a posición absoluta
K=5	Trazar línea hasta posición absoluta en color de primer plano
K=6	Trazar línea hasta posición absoluta en color lógico inverso
K=7	Trazar línea hasta posición absoluta en color de fondo

PLOT1 y PLOT5 son equivalentes a la utilización de las instrucciones MOVE y DRAW. Para nuestro efecto de explosión usaremos PLOT1 para dibujar líneas relativas al centro de la explosión. Sin embargo, antes de dibujar cualquier línea es necesario saber dónde está el centro de la explosión.

En el último capítulo del curso, cuando llamamos a una versión ficticia de PROCexplosión, les pasamos a x-explosión e y-explosión los valores de xgraf e ygraf. Éstos no jugaban ningún papel en el procedimiento ficticio, pero ahora los utilizaremos para especificar el centro de la explosión. Si se llama a este procedimiento desde la línea 3390 de PROCmover (véase p. 915), entonces xgraf e ygraf serán las coordenadas de gráficos del centro de la celda de carácter que contiene la mina. Por tanto, MOVExexplosión,y-explosión desplazará el cursor para gráficos hasta el centro del lugar en el que deseamos que se produzca la explosión. El motivo por el cual pasamos las coordenadas a otro par de variables para su empleo en el procedimiento de la explosión, en vez de utilizar simplemente xgraf e ygraf, es que este procedimiento también será llamado desde otra parte del programa en la cual las coordenadas del centro de la explosión se especificarán mediante un par diferente de variables. Pasando parámetros al procedimiento de la explosión lo hacemos más flexible y general.

Después de habernos desplazado (MOVE) hasta el centro de la explosión, trazaremos una línea en una dirección escogida al azar de, por ejemplo, un máximo de 50 unidades:

```
PLOT1,RND(50),RND(50)
```

no realizará del todo esta tarea, porque no especifica coordenadas negativas. (Una coordenada *x* negativa produce una línea trazada hacia la izquierda, mientras que una coordenada *y* negativa da una línea trazada hacia abajo desde el centro de la explosión.) El uso repetido de esta instrucción rellenará sólo la cuarta parte del espacio alrededor del centro (donde ambas coordenadas son positivas).

Para que podamos introducir valores negativos (y dibujar entonces líneas en todas las direcciones alrededor del centro de la explosión) debemos reemplazar RND(50) por RND(100)−50. El valor máximo de RND(100) es, por supuesto, 100, de modo que el valor máximo que puede tener

RND(100)−50 es 100−50=50. Por otra parte, el valor mínimo de RND(100) es uno. Así, RND(100) puede ser tan pequeño como 1−50=−49.

Mediante un movimiento (MOVE) repetido hacia el centro de la explosión y trazando una línea relativa de hasta 50 unidades en cualquier dirección y color podemos producir un efecto visual atractivo.

Consideremos ahora cómo podemos producir efectos de sonido para acompañar los gráficos explosivos de la pantalla. Se pueden crear sonidos bastante sofisticados utilizando las instrucciones SOUND y ENVELOPE del BASIC BBC. SOUND puede generar notas con tonos (de modo que se pueda tocar una melodía) o bien "ruido" para efectos especiales de sonido. La instrucción ENVELOPE se emplea junto con SOUND para "dar forma" al sonido que uno oye y simular, por tanto, instrumentos musicales. Hay cuatro números (o parámetros) relacionados con SOUND que controlan las características del tono producido. Éstos son los siguientes:

```
SOUND C,A,P,D
```

- C es el número de canal, que debe tener un valor entre 0 y 3. Los canales del 1 al 3 producen notas con tono, pero el canal 0 es para efectos especiales. Éste es el canal que vamos a utilizar aquí.
- A es la amplitud o, para decirlo de forma más sencilla, el volumen de sonido producido. El volumen máximo se establece cuando A es −15 y pasa a silencio cuando A es cero. Los valores positivos de A se utilizan para seleccionar la forma del sonido y se emplean junto con la instrucción ENVELOPE.
- P suele ser el tono de la nota; en otras palabras, lo agudo o grave que es el sonido.
- D es la duración o período de tiempo durante el cual suena la nota. Los valores de D de 0 a 254 hacen que la nota se mantenga durante un tiempo medido en unidades de una vigésima de segundo. Por consiguiente, para mantener una nota durante un segundo el valor de D se ha de establecer en veinte. Si el valor de D es −1, entonces el sonido continuará hasta que uno emprenda alguna acción para interrumpirlo.

El efecto sonoro de la explosión necesita tener un volumen máximo, de modo que nuestra instrucción tendrá la amplitud máxima, representada mediante un valor de −15. El valor del tono es algo más complicado. Cuando se utiliza con el canal 0, P toma valores entre 0 y 7 de acuerdo a estas reglas:

P=0	Vibración de alta frecuencia
P=1	Vibración de frecuencia media
P=2	Vibración de baja frecuencia
P=3	Vibración, frecuencia establecida por canal 1
P=4	Ruido de alta frecuencia
P=5	Ruido de frecuencia media
P=6	Ruido de baja frecuencia
P=7	Ruido, frecuencia establecida por canal 1

El grupo de cuatro formas de onda de vibración produce sonidos tipo "zumbador", mientras que las formas de onda de ruido producen algo que suena como un televisor antes de sintonizarlo adecuadamente. La instrucción SOUND que utilizaremos en nuestro programa es:

```
SOUND 0,−15,4,40
```

que produce un silbido agudo y dura dos segundos (40/20) con el volumen establecido en el máximo. He aquí el listado completo para el procedimiento:

```
3550 DEF PROCexplotar(x−explosion,y−explosion)
3560 REM ** EFECTO SONORO **
3570 SOUND 0,−15,6,50
3580 REM ** ESTABLECER VELOCIDAD FLASH **
3590 *FX9,20
3600 *FX10,50
3610 FOR I=1 TO 100
3620 MOVE x−explosion,y−explosion
3630 VDU19,2,RND(15),0,0,0
3640 GCOL 0,RND(3)
3650 PLOT 1,RND(100)−50,RND(100)−50
3660 NEXT I
3670 PROCrestaurar
3680 ENDPROC
```

El procedimiento "restaurar"

Tras una explosión hay varias tareas que realizar:

- Debemos reducir la cantidad de vidas que quedan y verificar si se han utilizado ya todas ellas.
- Debemos poner en orden todo el lío que se ha producido en la pantalla a resultas de la explosión.
- Debemos reposicionar el detector y el ayudante en su posición de comienzo.

La primera tarea se consigue mediante el empleo de un contador que se incrementa cada vez que se llama al procedimiento. Si el contador, después del aumento, es mayor de cuatro, se pone a uno una variable especial, flag final (indicador de final), para señalar que el juego ha concluido.

Habiendo verificado si el juego ha terminado, debemos limpiar la explosión. Para esto se podrían utilizar diversos métodos, como imprimir (PRINT) espacios en blanco encima de la superficie de la explosión. No obstante, el procedimiento que empleamos es limpiar la pantalla, volver a colocar las minas y reimprimir la información relativa a tiempo y marcador. Para mantenernos ecuánimes, deberíamos volver a colocar la misma cantidad de minas que quedaba antes de la explosión. Esto se puede calcular a partir del marcador. Dado que cada mina vale 150 puntos y que originalmente hay 50, podemos calcular la cantidad de minas que quedan y pasarla al procedimiento "colocar minas".

El detector de minas y el ayudante se vuelven a posicionar reinicializando sus coordenadas (llamando al procedimiento inicializar variables) y llamando luego al procedimiento situar sujetos. Abajo se ha listado el procedimiento restaurar completo. Agregue este listado y el procedimiento explosión.

```
3880 DEF PROCrestaurar
3890 contador=contador+1
3900 IF contador > 4 THEN flag−final=1:ENDPROC
3910 CLS
3920 VDU19,2,2,0,0,0
3930 COLOUR 2
3940 PROCinicializar−variables
3950 minas−restantes=50−marcador/150
3960 PROCcolocar−minas(minas−restantes)
3970 PROCtrazar−borde
3980 PRINTTAB(2,27);"Tiempo"
3990 PRINTTAB(2,28)"Marcador"
4000 PRINTTAB(11,28)marcador$
4010 PRINTTAB(2,29)"Max marcador"
4020 PRINTTAB(11,29)max−marcador$
4030 vidas−restantes$=LEFT$(hombres$,4−contador)
4040 COLOUR 1
4050 PRINTTAB(2,30);vidas−restantes$;"   "
4060 COLOUR 2
4070 PROCsituar−sujetos
4080 ENDPROC
```

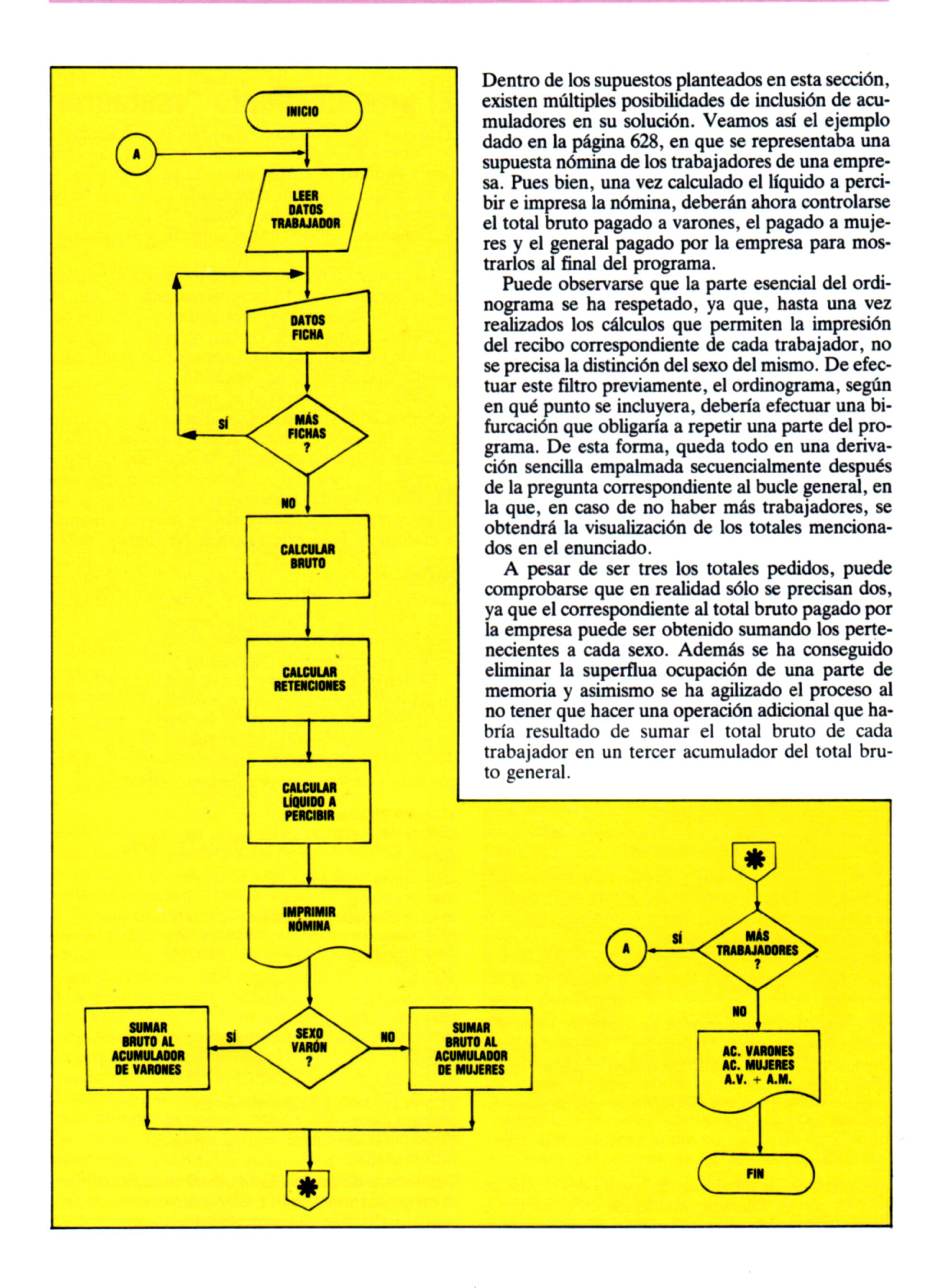
También es necesaria la siguiente modificación en el programa temporal de llamada (véase p. 885):

```
60 UNTIL TIME > 12099 OR flag final=1
```

Kevin Jones

Uso de acumuladores

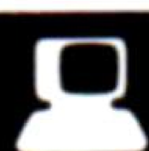
Puede ser muy práctico incluir acumuladores en la resolución de un problema representado en un diagrama de flujo



Dentro de los supuestos planteados en esta sección, existen múltiples posibilidades de inclusión de acumuladores en su solución. Veamos así el ejemplo dado en la página 628, en que se representaba una supuesta nómina de los trabajadores de una empresa. Pues bien, una vez calculado el líquido a percibir e impresa la nómina, deberán ahora controlarse el total bruto pagado a varones, el pagado a mujeres y el general pagado por la empresa para mostrarlos al final del programa.

Puede observarse que la parte esencial del ordinograma se ha respetado, ya que, hasta una vez realizados los cálculos que permiten la impresión del recibo correspondiente de cada trabajador, no se precisa la distinción del sexo del mismo. De efectuar este filtro previamente, el ordinograma, según en qué punto se incluyera, debería efectuar una bifurcación que obligaría a repetir una parte del programa. De esta forma, queda todo en una derivación sencilla empalmada secuencialmente después de la pregunta correspondiente al bucle general, en la que, en caso de no haber más trabajadores, se obtendrá la visualización de los totales mencionados en el enunciado.

A pesar de ser tres los totales pedidos, puede comprobarse que en realidad sólo se precisan dos, ya que el correspondiente al total bruto pagado por la empresa puede ser obtenido sumando los pertenecientes a cada sexo. Además se ha conseguido eliminar la superflua ocupación de una parte de memoria y asimismo se ha agilizado el proceso al no tener que hacer una operación adicional que habría resultado de sumar el total bruto de cada trabajador en un tercer acumulador del total bruto general.



Unidad transformadora

La nueva interface Electron Plus 1 puede convertir el Acorn Electron estándar en una alternativa económica del BBC Micro

El Acorn Electron se concibió como una versión a escala reducida del BBC Micro. Sus diseñadores conservaron el excelente BASIC BBC y su sistema operativo, pero eliminaron la mayoría de las interfaces que hacen del BBC Micro una máquina tan versátil. De hecho, las únicas conexiones del Electron son una puerta para cassette, dos conectores para pantalla (RGB o video compuesto), una salida de modulador para conectar a un televisor y un conector para la toma de corriente.

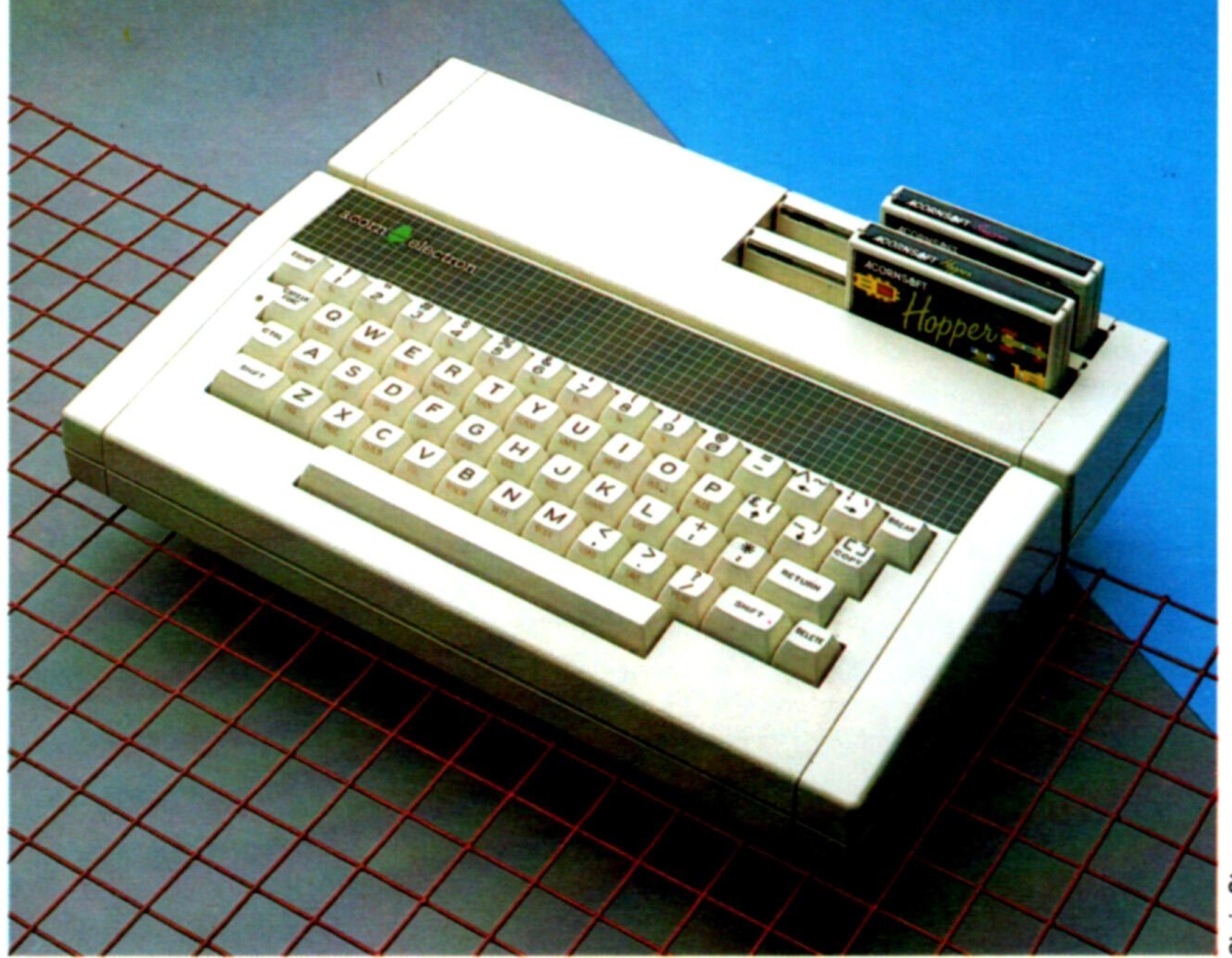
La interface Plus 1 le añade a la máquina estándar una puerta para impresora en paralelo, un conector para palanca de mando y dos conectores para cartuchos de ROM enchufables. Acorn espera que las facilidades extras aumenten el potencial de ventas de la máquina. Las ranuras para cartuchos y la puerta para palanca de mando permiten emplear el Electron en juegos de tipo recreativo, mientras que la puerta para impresora es una facilidad útil para aquel usuario que necesite imprimir listados de programas o que desee utilizar software para tratamiento de textos.

La instalación de la unidad de ampliación Plus 1 es muy sencilla: se acopla simplemente en el conector marginal que sobresale de la parte posterior del Electron. Luego se cierran dos pestillos sobre la carcasa del Plus 1 y se fijan a los conectores de rosca de la carcasa del ordenador. Tanto la corriente como los datos se transfieren a través del conector marginal. El software adicional necesario para "activar" la nueva interface está contenido en un chip de ROM dentro de la unidad.

La puerta para impresora es del tipo Centronics en paralelo estándar. Para habilitar la impresora, el usuario ha de entrar la instrucción VDU2, y para inhabilitarla debe entrar VDU3. Estas instrucciones son idénticas a las que se utilizan en el BBC Micro. Igualmente, se emplea VDU1 para enviarle un carácter a la impresora; esto también es estándar en el BASIC BBC.

Las palancas de mando se enchufan en la puerta analógica. Ésta es un conector tipo D de 15 patillas, y las conexiones de las patillas son idénticas a las de la puerta analógica del BBC Micro, lo que permite utilizar cualquier palanca de mando compatible con esta máquina. El Plus 1 puede medir hasta cuatro voltajes analógicos, ya sea provenientes de cuatro paletas para juegos o de dos palancas de mando (cada palanca produce dos voltajes: uno para el movimiento arriba-abajo y otro para izquierda y derecha).

El proceso de conversión de voltajes analógicos a señales digitales que puede tratar el ordenador se lleva a cabo mediante un chip convertidor de analógico a digital; en este caso, un ADC0844. Este chip es menos complejo que el que se utiliza en el BBC Micro, lo que significa que el Electron no sigue el movimiento de una palanca de mando con la misma precisión que lo hace una máquina BBC. En los programas de juegos, en los que la palanca se emplea simplemente como un mando de cuatro direcciones, ello no constituye un problema. Si se emplea la palanca de mando con un programa para gráficos (para dibujar "a pulso" en la pantalla, p. ej.), en el Electron los resultados son bastante inferiores y la imagen resultante menos definida.



Chris Stevens

Notable ampliación
El Electron se concibió originalmente como una versión de precio reducido del BBC Micro. Carece de casi todas las interfaces del BBC, a pesar de lo cual ofrece los mismos excelentes gráficos y BASIC a la mitad de precio. Ahora Acorn ha producido una unidad denominada Plus 1, que le proporciona al Electron las interfaces más importantes

Cuando se enciende por primera vez la Electron Plus 1, el cartucho del conector de ROM frontal comienza a funcionar automáticamente; para detenerlo se debe pulsar Escape. El sistema de archivo de ROM trabaja de forma muy similar a la versión en cassette (si bien es mucho más rápido), por lo cual interpreta instrucciones como *CAT, LOAD y CHAIN. Para utilizar software en cassette con la nueva unidad, hay que entrar la instrucción *TAPE o quitar el cartucho, recordando desconectarlo de la red previamente. Además de programas de juegos, también se suministran otros lenguajes soportados en cassette. A diferencia de las ROM de juegos, estas ROM de lenguajes se "encienden" para sustituir a la ROM de BASIC usual.

El problema de la mayoría de los accesorios para ordenador es que suelen tener efectos colaterales. Cuando se le conecta al Electron la unidad Plus 1, se pueden producir errores de carga al emplear programas en cassette. Esto afecta especialmente a los programas que contienen archivos de datos; por nuestra parte hemos descubierto que dos de los programas de la cassette de presentación del Electron no se cargan en el Electron ampliado. No obstante, utilizando una serie de instrucciones del sis-





Seis títulos de software
Los cartuchos para el Electron tienen una ventaja sobre la cinta, y es que cargarlos apenas lleva un par de segundos, frente a los varios minutos que lleva la carga de una cinta. En formato de cartucho sólo hay seis títulos a la venta. Éstos son cuatro juegos (entre ellos *Hopper* y *Snapper*), un programa educativo y el lenguaje LISP

tema operativo, uno puede "inducir" al Electron a pensar que no tiene conectada la Plus 1 y, por consiguiente, restaurar su rendimiento normal. En el manual esto casi ni se menciona y puede resultarle confuso al usuario novato.

En cualquier máquina Acorn, teclear *HELP proporciona una lista de las diversas ROM que hay en el ordenador. Tecleándola en el Electron con la unidad Plus 1 instalada, le dice que la principal ROM del sistema operativo es OS 1.00, y también lista Expansion 1.00 ADC/Printer/RS423. Ésta es la ROM del sistema operativo que hay en el interior de la unidad de ampliación Plus 1, pero lo interesante de esto es la última parte de la información: RS423. Ésta es una referencia a una interface en serie estándar, si bien ni el Electron ni la Plus 1 admiten dicha facilidad. De hecho, Acorn tiene en sus planes introducir una interface en serie más adelante.

La principal diferencia entre los dos BASIC es que el Electron carece de la modalidad 7, la modalidad de gráficos compatible con teletexto. Esta modalidad se utiliza para títulos y páginas de instrucciones en los programas para el BBC Micro, porque es económica desde el punto de vista de la memoria. En el Electron, los diversos atributos de videotexto no encienden los colores ni las letras intermitentes de la modalidad 7; en cambio, se visualizan en la pantalla como un galimatías. La mayoría de los programas para juegos utilizan para el juego propiamente dicho una modalidad diferente, de modo que después de las páginas de título e instrucciones ya no existe ningún problema.

La otra diferencia fundamental que existe entre las dos máquinas reside en las instrucciones SOUND y ENVELOPE. El Electron posee un único canal de sonido, frente a los cuatro que ofrece el BBC. Del mismo modo, la instrucción ENVELOPE del Electron sólo afecta al tono y no al volumen. No obstante, las instrucciones son compatibles, de modo que un programa que las utilice funcionará en ambas máquinas, si bien en el Electron el sonido será sensiblemente distinto.

El teclado del Electron tiene mejor "tacto" que el del BBC Micro, pero posee menos teclas. Ello significa que en el Electron, al igual que en el Sinclair Spectrum, la mayoría de las teclas tienen tres o cuatro funciones diferentes. Por ejemplo, la tecla "L" producirá "l" o "L", según se pulse o no la

Cartuchos
Acornsoft produce varios programas en formato de cartucho para utilizar con el Electron

Ranuras de ampliación
Estas dos tapas se articulan hacia abajo para posibilitar la inserción de cartuchos y otros accesorios en las dos ranuras de ampliación. Acorn tiene previsto producir una interface RS232 que utilizará estas ranuras, y también es probable que otras empresas independientes produzcan accesorios para ellas

Interface Centronics
Permite que el Electron utilice los tipos más populares de impresoras

Puerta de A a D
Esta puerta posee cuatro canales separados que llevan a un convertidor de analógico a digital (de A a D). Éste se puede utilizar para medir voltajes pequeños, aunque con menos precisión que el del BBC Micro. Se la suele emplear para instalar una o dos palancas de mando

tecla de cambio (Shift); pero si se utiliza junto con la tecla de función (Function), la tecla "L" generará la instrucción LIST del BASIC. Pulsando la tecla de control (Control) al mismo tiempo que "L" se limpiará la pantalla. Quizá hubiese sido preferible que Acorn hubiera conservado las 10 teclas de función rojas que se hallan en el BBC Micro, porque además se pueden programar para realizar diversas tareas útiles. El Electron posee teclas de función, pero éstas son sólo las teclas numéricas que se pulsan conjuntamente con la tecla Function.

El único aspecto en el que las dos máquinas difieren de forma notable es en la gama de interfaces proporcionadas. El BBC Micro tiene más interfaces que ningún otro ordenador personal, lo que significa que se le puede acoplar fácilmente un sinnúmero de periféricos, desde impresoras, modems y unidades de disco hasta segundos procesadores y



Juego rápido
El Snapper, de Acornsoft, es una buena versión del clásico juego Pac-Man. Se vende en formato de cartucho y de cassette y se puede controlar mediante teclas o con una palanca de mando

nector para ampliación
Plus 1 se conecta con el ctron a través de este nector marginal. Éste es el ico medio de ampliación de e dispone el Electron

brazos-robot. Todos éstos requieren chips y conectores adicionales, lo que inevitablemente aumenta los costos de fabricación. Omitiendo estas interfaces, no sólo ya no son necesarios chips extras, sino que los requerimientos de potencia son inferiores y el ordenador es más pequeño y más económico. Ésa fue precisamente la filosofía que inspiró la creación del Electron original. El hecho de que Acorn haya producido la Plus 1 sugiere que quizás este recorte del costo haya ido demasiado lejos.

El Electron Plus 1 es menos versátil que el BBC Micro y ofrece menos posibilidades de ampliación. Pero muchos usuarios no necesitan tales facilidades y, a un precio considerablemente inferior que la máquina BBC, el Electron Plus 1, con sus excelentes gráficos y facilidades de sonido, su BASIC estructurado y una creciente gama de software, parece ser una inversión rentable.

BBC MICRO MODELO B

DIMENSIONES
75 × 340 × 410 mm

CPU
6502, 1,8 MHz

MEMORIA
32 K de RAM, 32 K de ROM

PANTALLA
Ocho modalidades de visualización. Resolución más alta: texto, 80 × 32 caracteres; gráficos, 640 × 256 pixels. Hasta ocho colores que pueden ser fijos e intermitentes. Modalidad de visualización en teletexto. Caracteres definibles por el usuario

INTERFACES
UHF para televisión; RGB y video compuesto para pantallas; puerta para cassette; interfaces para impresoras RS423 y Centronics; puerta analógica (para palanca de mando, etc.); conectores de ROM (para software); capacidad para segundos procesadores; bus de 1 MHz; interface Disk (opcional); interface Econet para conexión en red (opcional); puerta para el usuario; salidas de corriente auxiliares (para unidades de disco, etc.)

LENGUAJES DISPONIBLES
BASIC BBC (incluido), Assembler 6502 (incluido), LISP, FORTH, BCPL, PASCAL

TECLADO
72 teclas tipo máquina de escribir. Incluye 10 teclas de función programables

DOCUMENTACION
El manual del BBC es una guía excelente para el programador experimentado, pero es de poca ayuda para el usuario principiante

VENTAJAS
Una de las mejores versiones de BASIC existentes, una enorme gama de interfaces, buenos gráficos, sonido y teclado

DESVENTAJAS
Iniciarse con el BBC no es fácil para el principiante



ACORN ELECTRON PLUS 1

DIMENSIONES
65 × 260 × 340 mm (con la Plus 1 instalada)

CPU
6502, 1,8 MHz

MEMORIA
32 K de RAM, 32 K de ROM

PANTALLA
Siete modalidades de visualización. Resolución más alta: texto, 80 × 32 caracteres; gráficos, 640 × 256 pixels. Hasta ocho colores y ocho intermitentes. Caracteres definibles por el usuario

INTERFACES
Salida UHF para televisión; RGB y video compuesto para pantallas; puerta para cassette; puerta para impresora en paralelo; puerta analógica (para palancas de mando, etc.); dos conectores (para software en cartucho de ROM, etc.)

LENGUAJES DISPONIBLES
BASIC, Assembler 6502 (suministrado); FORTH, LISP (también en cartucho de ROM), S-PASCAL

TECLADO
56 teclas tipo máquina de escribir. Tecla de función que permite la entrada de instrucciones de BASIC con una única tecla. Diez teclas programables

DOCUMENTACION
La Guía para el Usuario está bien realizada y es de fácil lectura. Posee una cobertura muy profunda del BASIC Electron y del lenguaje Assembly 6502

VENTAJAS
Los gráficos son buenos; producen una imagen clara. Teclado de gran calidad. Una buena versión de BASIC

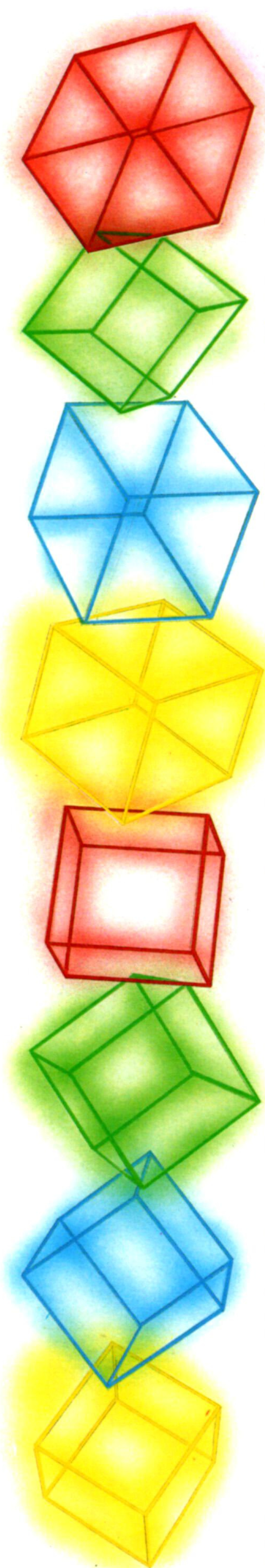
DESVENTAJAS
Las teclas multifunción pueden ser fuente de confusión. Sólo dispone de un canal de sonido. Poca memoria disponible. Carece de modalidad de teletexto.



Chris Stevens

Revolución cubista

Analizamos hoy un programa que nos permite dibujar formas tridimensionales y hacerlas rotar



El programa que proporcionamos utiliza algunos principios geométricos básicos para crear proyecciones en perspectiva de objetos que se pueden contemplar desde cualquier ángulo o distancia. Todos los datos para los objetos están almacenados en sentencias DATA dentro del programa. Éstos comprenden coordenadas tridimensionales para cada punto final de una línea del objeto. Para cada punto hay, asimismo, una cifra que indica si está al comienzo de una línea o en su final, que nos informa si la línea se ha de trazar desde o hacia dicho punto.

La conversión de estas coordenadas tridimensionales en valores bidimensionales que representen puntos en la pantalla es una cuestión simple, aunque extensa, y puramente matemática. Las coordenadas *x* e *y* de cada punto (en el plano de la pantalla de televisión) se dividen por un factor que representa la distancia del objeto en relación a nosotros. Además, la coordenada resultante se escala en función de un factor apropiado para el sistema de coordenadas del micro. Cambiando el factor distancia, se puede lograr que el objeto se encoja o agrande a medida que se acerca o se aleja del observador.

Se puede utilizar una tercera constante para cambiar el efecto de la proyección en perspectiva. Incrementando el valor de esta constante, se exagera la vista en perspectiva del objeto, como si se estuviera contemplando con una lente de ojo de pez. La disminución de esta constante da el efecto de aplanar la vista, como si estuviéramos mirando a través de un teleobjetivo.

Además de crear una vista en perspectiva del objeto, el programa también permite hacerlo rotar de modo que se lo pueda contemplar desde cualquier ángulo. Ello se consigue mediante simple trigonometría. Los ejes giran el ángulo deseado de forma que cuando se efectúa la proyección en perspectiva parece que la imagen de la pantalla ha girado. Esto se puede hacer bien como una rotación a través del eje *y* (el punto de vista parece moverse alrededor del objeto) o bien alrededor del eje *x* (el punto de vista se desplaza por arriba o por debajo del objeto).

Aunque nuestro programa puede crear todos estos efectos, su operatoria es notablemente sencilla. El control del punto de vista se ejerce mediante las teclas numéricas del uno al ocho. Éstas proporcionarán respectivamente: un movimiento del punto de vista hacia la izquierda, derecha, arriba, abajo, hacia el objeto o alejándose del mismo; un aumento del efecto de perspectiva (ojo de pez), o una disminución del efecto de perspectiva (teleobjetivo).

Las coordenadas tridimensionales corrientes se almacenan en tres matrices: X, Y y Z. Los cambios en estas matrices y los cambios en las constantes utilizadas para la transformación de la perspectiva se llevan a cabo en una serie de subrutinas. Cada vez que se detecta una pulsación de tecla, se borra la imagen de la pantalla dibujándola en el color de fondo, se realiza el cambio deseado con una llamada a la subrutina apropiada y se vuelve a dibujar la nueva imagen.

La transformación de la perspectiva se realiza en la subrutina que dibuja la imagen. Ésta pasa por cada juego de coordenadas tridimensionales, las traduce a bidimensionales y las traza en la pantalla (desplazándose a los puntos o dibujando líneas de acuerdo al cuarto dato para cada punto).

Crearse sus propios datos de objetos para este programa es una cuestión bastante larga pero relativamente fácil. Los datos se almacenan en sentencias al final del programa, con cuatro valores para cada punto del objeto. El número total de puntos se establece en la primera línea del programa. Ésta habrá de modificarse para adaptarla a los datos propios de cada caso. Los datos que proporcionamos crean un cubo con una línea en diagonal a través de una cara.

Cada juego de cuatro valores está por este orden: valor trazar o dibujar, coordenada *x*, coordenada *y*, coordenada *z*. Los valores se calculan siguiendo mentalmente el boceto del objeto en tres dimensiones. Utilizando un lápiz imaginario, visite cada vértice del esquema del objeto uno a uno. Si su lápiz se desplaza hasta un punto sin dibujar una línea, se utiliza 4 para el primer valor. Un valor de 5 indica que se ha de trazar una línea hasta el punto desde el punto precedente. Se utilizan estos valores en particular porque hacen que el programa para el BBC Micro resulte menos complicado.

El origen de las coordenadas (0,0) está en el centro de la pantalla. Lo mejor es hacer que éste también sea el centro de su objeto. El eje *x* es el eje horizontal, con valores positivos yendo hacia arriba. El eje *z* es el eje hacia dentro y hacia fuera de la pantalla. La dirección positiva de este eje es hacia dentro de la pantalla.

Mantenga los valores de X, Y y Z tan pequeños como sea razonablemente posible. La preparación inicial del efecto de perspectiva y la distancia del punto de vista deben tener en cuenta que la anchura de un objeto de alrededor de 10 ocupará toda la pantalla. Esto se podría modificar alterando el valor de escalado utilizado en la transformación de la perspectiva. Le sugerimos que primero experimente con formas simples. Mantenga reducida la cantidad de puntos. Cuando haya conseguido dominar la digitalización de objetos sólidos simples (una pirámide, un cubo) puede atreverse con otros más complicados.

Nuestro programa se podría ampliar aún más para proporcionar efectos adicionales. Se podría incorporar una facilidad de traslación para desplazar el objeto, sin rotación, en relación al origen de la



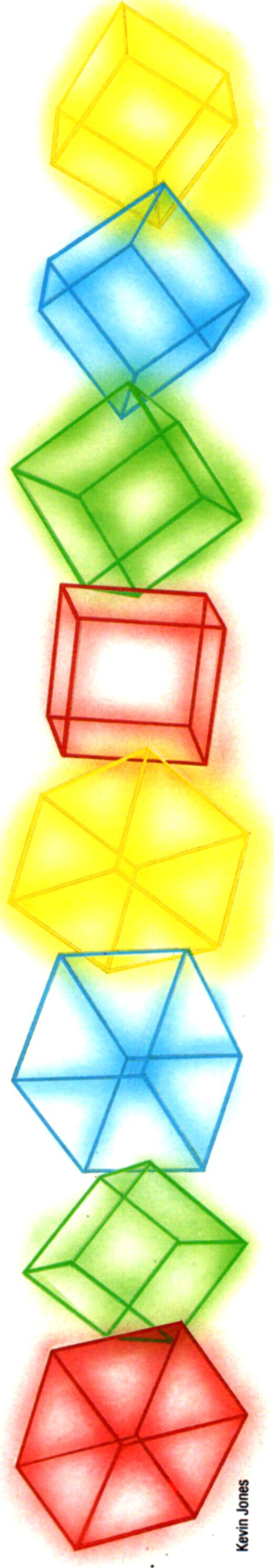
coordenada. Intente incorporar una rutina para eliminar las líneas y las porciones de líneas que quedarían ocultas a la vista. Con ello se lograría hacer que la actual imagen, como construida en alambre, tuviera un aspecto más realista. No obstante, la eliminación de estas líneas ocultas es un asunto enormemente complicado. Exige unas matemáticas muy complejas y haría que el programa resultara considerablemente más lento. Aun si no se le agrega nada al programa y se le deja tal como se ofrece aquí, se pueden conseguir algunos resultados atractivos y espectaculares.

Versión para el Spectrum

```
10 LET N=16
20 DIM P(50)
21 DIM X(50)
22 DIM Y(50)
23 DIM Z(50)
24 DIM A(50)
25 DIM B(50)
40 LET D=10: LET P=0.5
50 LET SI=SIN 0.09: LET CO=COS 0.09
60 FOR I=1 TO N
70 READ P(I),X(I),Y(I),Z(I)
80 NEXT I
90 :
200 INVERSE 0: GO SUB 300
210 IF INKEY$<>"" THEN GO TO 210
211 IF INKEY$="" THEN GO TO 211
212 LET I$=INKEY$
230 INVERSE 1: GO SUB 300
240 IF I$="1" THEN GO SUB 1000
241 IF I$="2" THEN GO SUB 2000
242 IF I$="3" THEN GO SUB 3000
243 IF I$="4" THEN GO SUB 4000
244 IF I$="5" THEN GO SUB 5000
245 IF I$="6" THEN GO SUB 6000
246 IF I$="7" THEN GO SUB 7000
247 IF I$="8" THEN GO SUB 8000
250 GO TO 200
260 :
300 FOR I=1 TO N
310 LET A(I)=X(I)*300/(P*Z(I)+D): LET B(I)=
    Y(I)*300/(P*Z(I)+D)
320 NEXT I
330 FOR I=1 TO N
340 IF P(I)=4 THEN PLOT A(I)+128,B(I)+85
345 IF P(I)=5 THEN DRAW A(I)-A(I-1),B(I)-B(I-1)
350 NEXT I
360 RETURN
370 :
1000 FOR I=1 TO N
1010 LET X=X(I)*CO-Z(I)*SI
1020 LET Z=Z(I)*CO+X(I)*SI
1030 LET X(I)=X: LET Z(I)=Z
1040 NEXT I
1050 RETURN
1060 :
2000 FOR I=1 TO N
2010 LET X=X(I)*CO+Z(I)*SI
2020 LET Z=Z(I)*CO-X(I)*SI
2030 LET X(I)=X: LET Z(I)=Z
2040 NEXT I
2050 RETURN
2060 :
3000 FOR I=1 TO N
3010 LET Y=Y(I)*CO+Z(I)*SI
3020 LET Z=Z(I)*CO-Y(I)*SI
3030 LET Y(I)=Y: LET Z(I)=Z
3040 NEXT I
3050 RETURN
3060 :
4000 FOR I=1 TO N
4010 LET Y=Y(I)*CO-Z(I)*SI
4020 LET Z=Z(I)*CO+Y(I)*SI
4030 LET Y(I)=Y: LET Z(I)=Z
4040 NEXT I
4050 RETURN
4060 :
5000 LET D=D*0.9
5010 RETURN
5020 :
6000 LET D=D/0.9
6010 RETURN
6020 :
7000 LET P=P/0.9
7010 RETURN
7020 :
8000 LET P=P*0.9
8010 RETURN
8020 :
9000 DATA 4,1,1,1, 5,1,1,-1, 5,-1,1,-1, 5,-1,1,1,
     5,1,1,1
9010 DATA 5,1,-1,1, 5,1,-1,-1, 5,-1,-1,-1, 5,-1,-1,1,
     5,1,-1,1
9020 DATA 4,1,-1,-1, 5,1,1,-1, 5,-1,-1,-1,
     5,-1,1,-1
9030 DATA 4,-1,1,1,5,-1,-1,1
```

Versión para el BBC

```
10 N=16
20 DIM P(50),X(50),Y(50),Z(50),A(50),B(50)
30 MODE0:VDU29,640;512;5
40 D=10:P=0.5
50 SI=SIN(0.09):CO=COS(0.09)
60 FOR I=1 TO N
70   READ P(I),X(I),Y(I),Z(I)
80   NEXT I
90 :
200 GCOL0,3:GOSUB 300
210 I$=GET$
220 V=VAL(I$)
230 GCOL0,0:GOSUB 300
240 ON V GOSUB 1000,2000,3000,4000,5000,6000,
    7000,8000 ELSE 250
250 GOTO 200
260 :
300 FOR I=1 TO N
310   A(I)=X(I)*1000/(P*Z(I)+D):B(I)=Y(I)*1000/(P*Z(I)+D)
320   NEXT I
330 FOR I=1 TO N
340   PLOT P(I),A(I),B(I)
350   NEXT I
360 RETURN
370 :
1000 FOR I=1 TO N
1010   X=X(I)*CO-Z(I)*SI
1020   Z=Z(I)*CO+X(I)*SI
1030   X(I)=X:Z(I)=Z
1040   NEXT I
1050 RETURN
1060 :
2000 FOR I=1 TO N
2010   X=X(I)*CO+Z(I)*SI
2020   Z=Z(I)*CO-X(I)*SI
2030   X(I)=X:Z(I)=Z
2040   NEXT I
2050 RETURN
2060 :
3000 FOR I=1 TO N
3010   Y=Y(I)*CO+Z(I)*SI
3020   Z=Z(I)*CO-Y(I)*SI
3030   Y(I)=Y:Z(I)=Z
3040   NEXT I
3050 RETURN
3060 :
4000 FOR I=1 TO N
4010   Y=Y(I)*CO-Z(I)*SI
4020   Z=Z(I)*CO+Y(I)*SI
4030   Y(I)=Y:Z(I)=Z
4040   NEXT I
4050 RETURN
4060 :
5000 D=D*.9
5010 RETURN
5020 :
6000 D=D/.9
6010 RETURN
6020 :
7000 P=P/.9
7010 RETURN
7020 :
8000 P=P*.9
8010 RETURN
8020 :
10000 DATA 4,1,1,1, 5,1,1,-1, 5,-1,1,-1, 5,-1,1,1,
      5,1,1,1
10010 DATA 5,1,-1,1, 5,1,-1,-1, 5,-1,-1,-1, 5,-1,-1,1,
      5,1,-1,1
10020 DATA 4,1,-1,-1, 5,1,1,-1, 5,-1,-1,-1, 5,-1,1,-1
10030 DATA 4,-1,1,1, 5,-1,-1,1
```



Kevin Jones

Estructuras modulares

En este capítulo mostramos cómo mejorar su programación utilizando módulos de código como componentes básicos

Un módulo es un trozo de código que realiza una función determinada. Los puntos de entrada y salida de los módulos, llamados *interfaces*, se deben definir con precisión, y los procesos que se producen entre estas interfaces deben ser absolutamente independientes del resto del programa. Una vez que se ha escrito un módulo, se lo puede tratar como a una "caja negra". Los datos puden entrar y salir del módulo a través de sus interfaces, pero lo que sucede dentro del mismo se puede dejar en sus propias manos.

Los módulos se pueden unir entre sí para construir un programa sin que el programador tenga que preocuparse por la forma en que llevan a cabo sus tareas. Un programador se puede construir un "stock" de módulos para emplearlos cuando los necesite y los programadores pueden intercambiar módulos para usarlos en sus programas. Pero para sacar partido de la programación estructurada modular, al escribir los módulos hay que tomar nota respecto al flujo de control y al flujo de datos.

Para asegurar que todos los módulos se comportan del mismo modo en relación al flujo de control, se debe seguir una regla muy simple: todos los módulos deben tener un único punto de entrada y un único punto de salida. Lo que esto significa en la práctica es que se debe diseñar cuidadosamente el flujo de control dentro del módulo de manera que empiece en un lugar e, independientemente de los bucles y las bifurcaciones en que pueda incurrir, llegue a la misma salida por todas las rutas posibles.

Los módulos corresponden a los algoritmos que hemos estado analizando en capítulos anteriores del curso. Los lenguajes "estructurados", como el PASCAL, permiten que el programador cree rutinas a las que se puede llamar por su nombre y que utilizan sus propias variables. Los lenguajes de esta naturaleza estimulan al programador para que entre y salga de una rutina (denominada *procedimiento*) por puntos únicos de entrada y salida.

En BASIC, utilizando la combinación GOSUB...RETURN se puede llamar a una subrutina desde el programa principal y, después de que la misma se haya ejecutado, el control retornará a la línea que sigue inmediatamente a la de la instrucción GOSUB. No obstante, no existe restricción respecto a la línea de la cual GOSUB le envía el control. Dos instrucciones GOSUB pueden enviar el control a líneas diferentes de una subrutina con un único RETURN, y el resultado puede ser completamente distinto en cada caso. Del mismo modo, no hay ninguna restricción respecto a cuántas sentencias RETURN se pueden emplear en una subrutina.

Todo ello significa que el programador de BASIC debe ser autodisciplinado. Debe empezar por asegurarse de que todas las GOSUB hacia la misma subrutina señalen al mismo número de línea y que todas las subrutinas posean en su interior un solo RETURN. Lo mejor es acostumbrarse a que la primera línea de cada subrutina contenga una sentencia REM que le otorgue un título y utilizar esa línea como punto de entrada. La última línea de la subrutina ha de ser el RETURN. Esto no es esencial, pero ayuda a que las cosas queden más claras.

La regla GOTO

Se debe tener cuidado especial con la instrucción GOTO, que puede hacer estragos en la estructura del programa. En este caso la regla es: utilizar un GOTO sólo para enviar el control a una línea *dentro* de la misma subrutina. Con ello se evita el peligro potencial de saltarse un RETURN o pasarle el control a un RETURN equivocado. En algunas ocasiones es necesario retornar de una subrutina sin ejecutar todas sus líneas. En este caso debe ir hacia (GOTO) la línea que posee el RETURN y de esta forma no se planteará ningún problema.

Utilizar instrucciones GOTO dentro de bucles es aún más peligroso. Si el control salta hacia fuera de un bucle, ¡el BASIC no puede saberlo y supone que el resto del programa es el cuerpo de ese bucle! La regla de seguridad es: estando en el cuerpo de un bucle, no ir nunca hacia (GOTO) una línea que esté fuera del cuerpo de ese bucle. Si necesita salir de un bucle antes de su fin, ponga el contador del bucle o la variable de condición en el valor terminal y vaya hacia (GOTO) la línea de condición (la línea que contiene NEXT o WHILE). Al igual que con la sentencia RETURN, coloque NEXT o WHILE en una línea exclusiva a efectos de claridad. Seguir la pista de la estructura de un programa es muchísimo más fácil si se evitan los GOTO en la medida de lo posible.

Es en las bifurcaciones donde es más probable que el control se extravíe, de modo que trate de que ninguna decisión envíe el control hacia fuera de una subrutina a menos que sea con una llamada correcta a otra subrutina. Recuerde que cada subrutina posee un único punto de salida, de modo que asegúrese de que sea posible seguir el flujo de control a través de todas las bifurcaciones hasta ese punto. Esto es más fácil de controlar si se dibuja un diagrama de flujo para la rutina. Con frecuencia, utilizar un flag o indicador puede reducir la necesidad de instrucciones GOTO en aquellas rutinas que implican bucles y bifurcaciones.

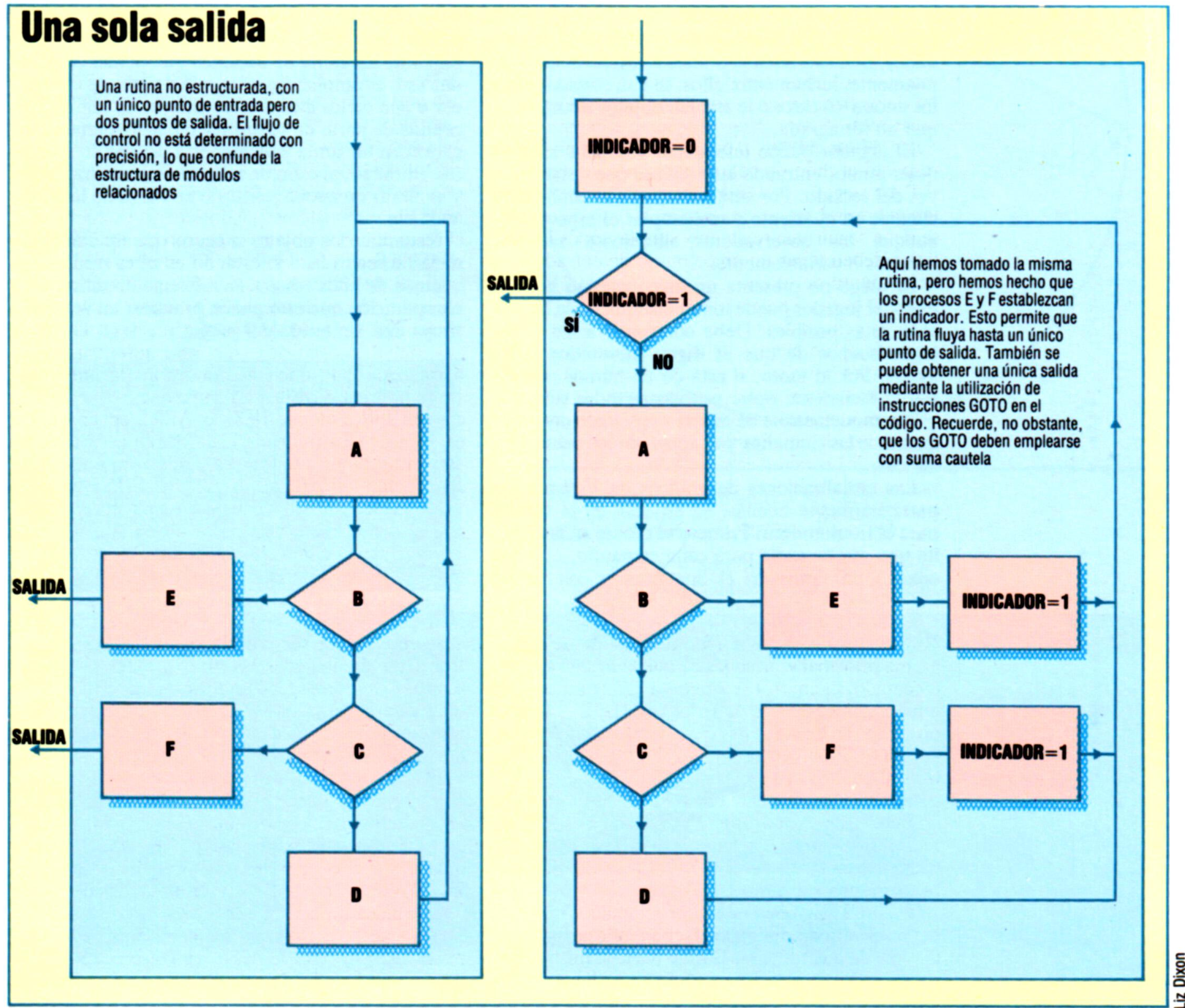
Podemos pensar que los datos entran y salen de los módulos, tal como aplicamos esta idea en el caso de los algoritmos (véase p. 866). Para que los módulos se puedan utilizar de forma independiente los unos de los otros, deben ser diseñados de modo que la única influencia que ejerzan entre sí sea en función de los datos que se pasan entre ellos. El programa principal le pasa datos a un módulo y, después de que éste se ha ejecutado, se retorna el resultado, sea cual sea.

Los datos se mueven por el programa dentro de variables y la libertad de movimiento de una variable se denomina su *ámbito*. Muchos lenguajes de programación pueden limitar el ámbito de una variable a determinadas subrutinas. En PASCAL, las variables que se utilizan en una determinada subrutina (procedimiento) se deben "declarar" para ese procedimiento. Las variables declaradas para el programa principal son *globales* y se podrían emplear en cualquier otro lugar del programa (incluyendo dentro de cualquiera de sus módulos). Sin embargo, las variables declaradas dentro de un determinado procedimiento son *locales* de ese procedimiento y sólo se pueden utilizar allí.

Las variables locales pueden tener el mismo nombre que las globales y la utilización de una no afecta al valor de la otra. El uso de un lenguaje que admita variables locales nos permite escribir subrutinas sin que debamos preocuparnos sobre cómo las variables empleadas en la rutina pueden afectar a las variables de otras subrutinas. Por desgracia, son pocas las versiones de BASIC que admiten variables locales, lo que significa que si deseamos escribir subrutinas independientes debemos simular de alguna forma el efecto de tener variables locales.

La forma más sencilla de simular este efecto consiste en adoptar convenciones de nomenclatura para distinguir a las variables que realizan tareas diferentes. Existen algunas convenciones ya aceptadas y los programadores se sirven de ellas ampliamente. Utilizar I, J y K como contadores de bucles y valores de índice es muy común y es una práctica que se ha adoptado a partir de las matemáticas.

Después de haber descrito un programa mediante un diagrama de flujo, es una cuestión sencilla numerar las subrutinas implicadas o proporcionarles algún otro tipo de código. Las variables globales que es necesario hacer locales para una subrutina determinada se pueden entonces dotar de un subíndice basado en este código para hacerlas exclusivas. Por consiguiente, la rutina número 5 podría utilizar las variables SUMA5 y TOTAL5 para distinguirlas de SUMA12 y TOTAL12 de la rutina número 12. No obstante, ¡tenga cuidado de que el BASIC que esté utilizando no mire sólo los dos primeros caracteres! No es necesario codificar las variables que se emplean para pasar valores entre subrutinas y aquellas que se utilizan sólo en el programa principal.



Jugando con dioses

"Valhalla" es un imaginativo juego de aventuras con constantes referencias a la mitología escandinava y espectaculares gráficos, escrito para el Spectrum y el Commodore 64

En la aventura *Valhalla* existen 81 escenarios, de los cuales 16 están en Asgard, incluyendo la evanescente Valhalla, el lugar soñado y buscado por el usuario. Midgard consta de 20 escenarios, y los 45 restantes componen colectivamente el Infierno. Cada escenario tiene ocho salidas, si bien algunas de ellas pueden estar bloqueadas o exigir la posesión de un objeto mágico para permitirle la entrada. Para complicar más las cosas, hay "caminos de anillos" que conectan escenarios alejados; si lleva un anillo adecuado puede "saltar" hasta el escenario más lejano.

Valhalla tiene un reparto de 36 personajes (en la pantalla habrá al menos tres en cualquier momento dado), que son buenos o malos. Interactúan continuamente, luchan entre ellos, se dan comida o vino los unos a los otros o le arrojan objetos a cualquiera que no les agrade.

El jugador puede interactuar con ellos en cualquier punto, entrando las acciones que desee a través del teclado. Por otra parte, puede también reclinarse en el asiento y contemplar el espectáculo, aunque los observadores silenciosos acabarán muertos invariablemente.

El juego no presenta un único camino hacia el éxito; el jugador puede tomar cualquiera de las muchas rutas posibles. Debe convencer a los personajes buenos de que es digno merecedor de su ayuda. Por lo tanto, si está de un humor pésimo, puede fácilmente optar por gastar todas sus energías comportándose de manera repelente para conquistarse las simpatías y el apoyo de los personajes malos.

Las visualizaciones de gráficos de *Valhalla* son particularmente buenas, en especial en la versión para el Commodore. Primero se dibuja en la pantalla una escenografía para cada escenario, en vivos colores primarios en el Spectrum y con suaves tonos pastel en el Commodore. A continuación aparecen los personajes: en el Spectrum éstos son estilizaciones con palitos y están dibujados en negro, mientras que en el Commodore, su mayor potencial de color y su mayor resolución permiten representar a los personajes con mayor detalle. Por último aparecen los objetos existentes en el escenario: éstos pueden incluir alimentos, vino, joyas, llaves y armas. Nuevamente, las representaciones del Spectrum son muchísimo menos realistas que las de la versión para el Commodore.

Los personajes interactúan, entonces, entre ellos y sus acciones se describen con palabras en la parte inferior de la pantalla. Y aquí es donde participa el jugador. La gama de acciones que puede emprender éste es amplia: quizá le guste tratar de convencer a uno de los dioses escandinavos de que se desprenda de parte de su tesoro, o puede intentar atacarlo con un arma para poner a prueba su fuerza. Sin embargo, al cabo de un rato deseará marcharse y explorar un nuevo territorio en busca de la mítica Valhalla.

Encontrar los objetos mágicos que finalmente le darán acceso a este Paraíso no es tarea fácil; coger algunos de ellos resulta increíblemente difícil, y la exasperación que esto puede provocar tal vez constituya una limitación del juego.

Valhalla: Para el Spectrum de 48 K y el Commodore 64
Editado por: Legend, Freepost, 1 Milton Road, Cambridge CB4 1UY
Autores: Graham Asher, Richard Edwards, Charles Goodwin, James Learmont, Jan Ostler, Andrew Owen, John Peel
Palancas de mando: No se necesitan
Formato: Cassette

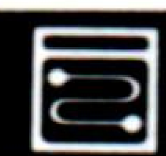
Día y noche

Valhalla es uno de los varios juegos que se han producido recientemente cuyos gráficos indican el paso del tiempo mediante el oscurecimiento de la pantalla. Ello crea el efecto de "día" y "noche"





Liz Heaney



En el redondel

Para el Commodore 64 ya explicamos varias subrutinas que explotaban sus posibilidades en alta resolución. La presente sirve para el trazado de círculos

No es posible dibujar con toda precisión una circunferencia en un ordenador familiar. El grado de precisión depende del método seguido y de la longitud y complejidad de la rutina resultante. El trazado por medio del BASIC por lo general se sirve de las funciones seno y coseno o de raíces cuadradas que proporcionan las coordenadas de los puntos sobre los que pasa la circunferencia. Ambos métodos tienen sus dificultades al tratar de adaptarlos a código máquina. Esto es lo que nos lleva a proponer un método alternativo, especialmente adecuado para su resolución en lenguaje máquina.

El método del que hablamos considera el diámetro de la circunferencia divisible en numerosas partes iguales, cada una de las cuales mide W. Por cada división debemos pensar en una línea que alcanza verticalmente un punto, C, de la circunferencia. La ilustración adjunta muestra una de estas líneas tras N divisiones del extremo izquierdo del diámetro AB. Uniendo A y B con C obtenemos dos triángulos rectángulos ACD y BCD, según puede verse.

Empleando el teorema de Pitágoras, podemos escribir las siguientes igualdades referidas al diagrama:

$$AC^2 = AD^2 + CD^2$$
$$CB^2 = DB^2 + CD^2$$

Sumando ambas igualdades obtenemos esta otra igualdad:

$$AC^2 + CB^2 = AD^2 + DB^2 + 2CD^2$$

Sabemos también por geometría que el triángulo ABC es igualmente rectángulo. Podemos, pues, establecer:

$$AC^2 + CB^2 = AB^2$$

Con lo cual la expresión que escribimos anteriormente queda así:

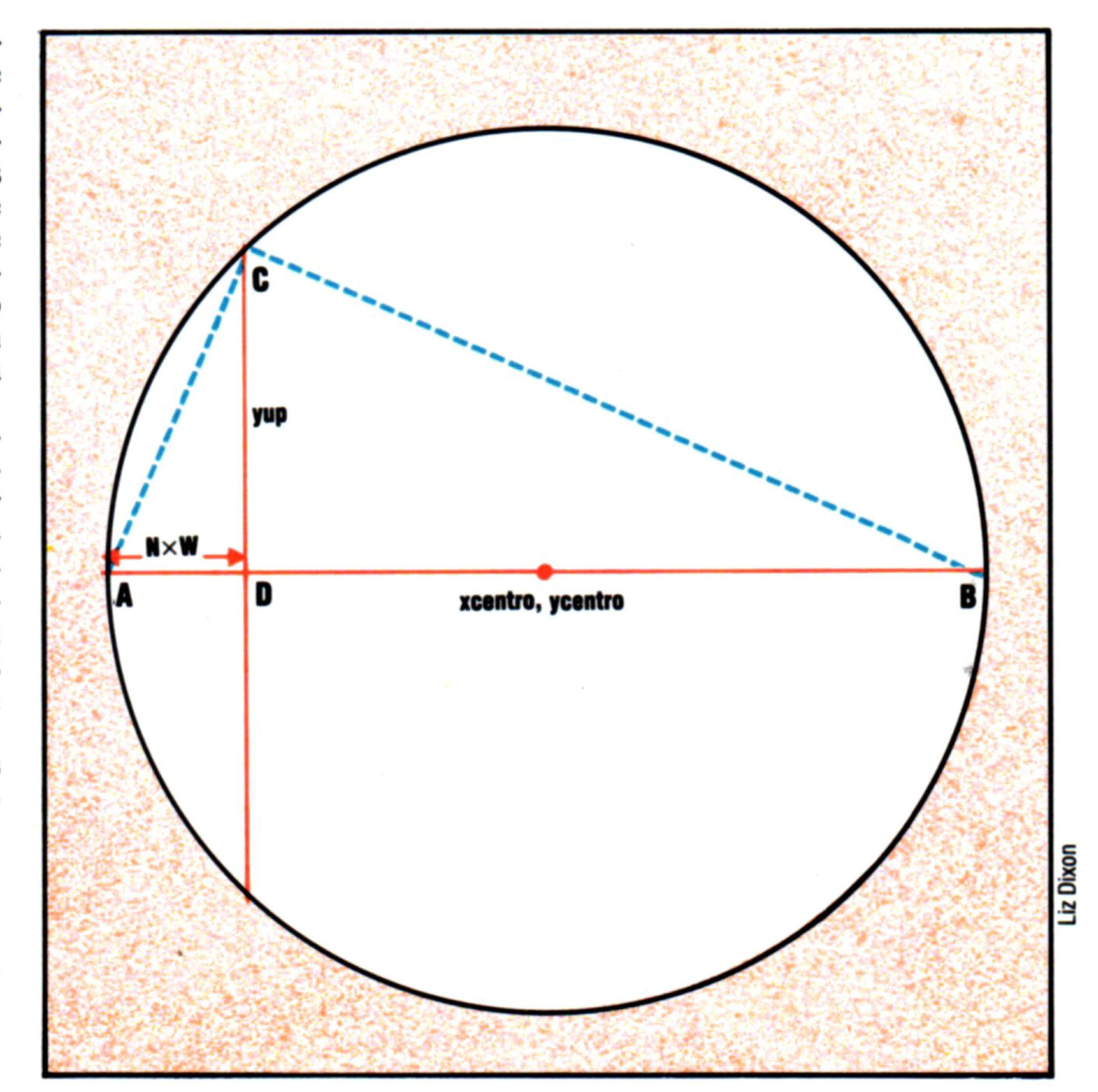
$$AB^2 = AD^2 + DB^2 + 2CD^2$$

A CD llamaremos "yup" (*y* "hacia arriba"), o sea, la distancia del punto del diámetro a la circunferencia. Pero AD vale N×W y AB mide 2×R (R es el radio). Sustituyendo en la ecuación anterior y despejando tendremos:

$$2yup^2 = (2R)^2 - (NW)^2 - (2R - NW)^2$$
$$yup^2 = 2RNW - (NW)^2$$

Si, por ejemplo, decidimos dividir el diámetro en 64 partes iguales, tendremos que W = 2×R/64, o lo que es lo mismo W = R/32. Volviendo a la ecuación:

$$yup^2 = 2RNR/32 - (NR)^2/32^2$$
$$yup^2=(64\ R^2N-R^2N^2)/32^2=R^2(64N-N^2)/32^2$$



Tomando raíz cuadrada en ambos miembros:

$$yup=\sqrt{R^2(64N-N^2)/32^2}=R/32\times\sqrt{64N-N^2}$$

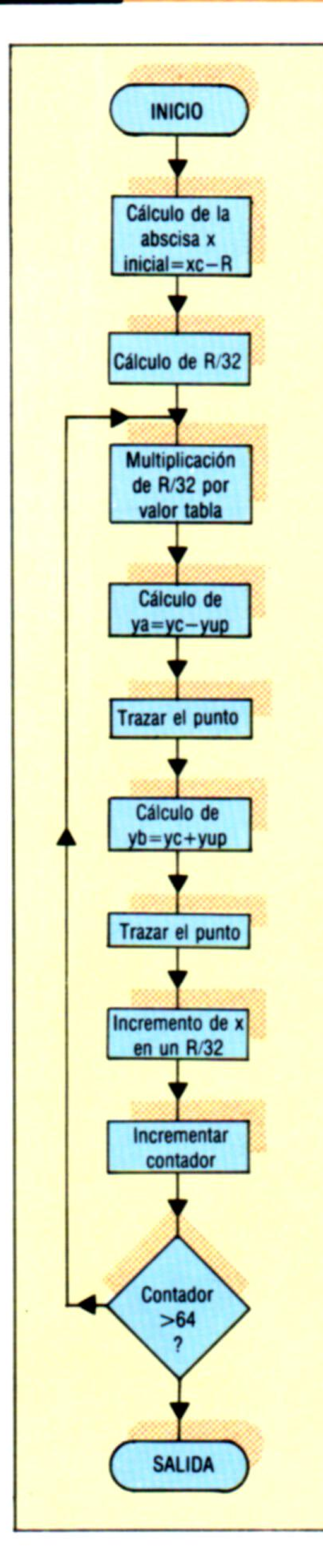
Si comenzamos por x = xcentro − R e incrementamos en 64 pasos iguales, entonces la distancia vertical hasta la circunferencia se obtiene con esa fórmula que acabamos de deducir, siendo N el número del incremento. Y aunque este resultado incluya una raíz cuadrada, la expresión a la que afecta la raíz no depende de las coordenadas del centro ni del radio. Podemos, pues, calcular una tabla de valores para la función raíz que nos proporcione una solución para cada uno de los valores de N, desde el 0 al 64. Esto se hace sólo una vez y se incorpora en el programa como tabla de consulta.

La ordenada absoluta *y* para cada incremento de *x* será:

$$ya = ycentro - yup$$

También podemos servirnos de la simetría de la circunferencia para calcular la ordenada *y* correspondiente a la otra mitad:

$$yb = ycentro + yup$$



Construcción de la rutina

Estamos en disposición de trazar las líneas maestras de nuestra rutina. El diagrama de flujo al margen muestra la manera de calcular cada punto de la circunferencia. Vemos que nuestra rutina, al necesitar tan sólo de una multiplicación, puede trazar cada punto con relativa rapidez. Pero hay dos ligeros inconvenientes. El primero es que no nos dará un círculo continuo sino una serie de puntos que componen una circunferencia. Y el segundo reside en que si bien esta técnica ofrece círculos bien definidos en BASIC, cuando se ejecuta en lenguaje máquina surgen imprecisiones.

El primer problema se resuelve utilizando la Linesub (subrutina de la línea; véase p. 899) para unir los puntos con pequeños segmentos y obtener así un círculo continuo. El segundo problema es debido a las inexactitudes en el cálculo de la raíz cuadrada que se ofrece en forma de tabla de consulta. El hecho de calcular los valores en BASIC y colocarlos (POKE) en una serie de bytes prefijados para la tabla significa que en realidad sólo queda almacenada la parte entera de cada valor. Si deseamos círculos mejor definidos debemos mejorar la precisión de los valores almacenados de la tabla. El valor máximo a almacenar es el 32; podemos, pues, multiplicar cada número por ocho antes de almacenarlo, sin dejar de utilizar un byte por cada entrada de la tabla.

El número 32 es el único que puede, multiplicado por ocho, ser guardado en un único byte. Para simplificar la rutina habremos de aproximar este valor. Así, nuestra rutina en código máquina puede dividir por ocho el valor de la tabla ejecutando tres LSR (*Logical Shifts Right*: desplazamientos lógicos a la derecha). Y lo que es aún más importante, el resto puede quedar guardado después de la división para ser utilizado en posteriores cálculos.

El listado del código fuente reserva 65 bytes al comienzo del programa para almacenar la tabla, etiquetando el primero de ellos. Las siguientes entradas de la tabla son así accesibles por medio del direccionamiento indexado. Este listado puede introducirse y ensamblarse en la memoria de la manera habitual. Aunque antes de guardar (SAVE) el código ensamblado debe introducirse y ejecutarse el siguiente programa en BASIC. Esto no dañará al código objeto, dado que se encuentra ubicado en la parte alta de la memoria. El programa calcula los 65 valores que necesita nuestra rutina Circsub (*subrutina Circular*), multiplica además estos valores por ocho y coloca (POKE) el resultado dentro del área de memoria reservada en el programa de código máquina.

Una vez ejecutado el programa de creación de la tabla, el código máquina puede guardarse (SAVE) del modo habitual, pero asegurando que se guarda (SAVE) la tabla junto con el código objeto. La tabla comienza en $C500. Una vez hecho esto, la tabla de consulta se cargará automáticamente siempre que carguemos Circsub.

```
 5 REM *** CREACION TABLA CIRCSUB ****
10 FOR N=0 TO 64
20 X=SQR (64*N−N↑2)*8
25 X %=X: DF=X−X %
27 IFRE >=.5 THEN X %=X %+1
28 IF X %>255 THEN X %=255
29 POKE 50432+N,X %
30 NEXT
```

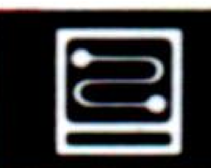
Este programa que ahora sigue muestra la manera como puede usarse desde un programa BASIC nuestra rutina Circsub. Sólo es necesario especificar las coordenadas del centro y el radio. En la línea 2000 la subrutina parte del valor de *x* en dos, según la forma byte *lo* byte *hi*; después coloca (POKE) en Circsub los valores especificados y hace la oportuna llamada (SYS). Observe que Circsub se vale de Linesub y esta subrutina a su vez de Plotsub (véase p. 817), por lo que las tres subrutinas han de cargarse al inicio del programa. Este programa dibuja círculos por la pantalla con radios cada vez mayores.

```
3 REM****PROGRAMA PRUEBA DE CIRCSUB****
5 DN=8:REM FOR CASSETTE DN=1
10 IFA=0THENA=1:LOAD"PLOTSUB.HEX",DN,1
15 IFA=1THENA=2:LOAD"LINESUB.HEX",DN,1
20 IFA=2THENA=3:LOAD"CIRCSUB.HEX",DN,1
100 GOSUB1000
110 YC=100:R=2
120 FORXC=30TO210STEP7
130 R=R+3
140 GOSUB2000
150 NEXT
160 GETA$:IFA$=""THEN160
170 GOSUB3000
180 END
1000 REM **** ACTIVACION ALT. RES. ****
1010 POKE49408,1:POKE49409,1
1020 POKE49410,3
1030 SYS49422
1040 RETURN
1050 :
2000 REM **** ENTRADA CIRCSUB ****
2010 CHI=INT(XC/256):CLO=XC−256*CHI
2020 POKE50497,CLO:POKE50498,CHI
2030 POKE50499,YC
2040 POKE50500,R
2050 SYS50521
2060 RETURN
2070 :
3000 REM **** BORRADO ALT. RES. ****
3005 RESTORE
3007 PRINTCHR$(147):REM CLEAR SCREEN
3008 PRINTTAB(10)"CIRCSUB VARIABLES"
3009 PRINT
3010 POKE49408,0:SYS49422
3030 FORI=50497TO50497+23STEP2
3035 READA$
3040 PRINTTAB(2);A$,PEEK(I),
3045 READA$
3047 PRINTA$,PEEK(I+1)
3050 NEXT
3060 RETURN
3070 :
5000 DATAXCTRLO,XCTRHI,YCTR,RADIUS,INTR
5010 DATAREMR,TOTX,RESREM,RESLO,RESHI,OLDXLO
5020 DATAOLDXHI,NEWXLO,NEWXHI,OLDYA
5030 DATAOLDYB,NEWYA,NEWYB,XFLAG,YFLAG,OLDY,NEWY
5040 DATAINTTAB,REMTAB
```

Como ocurre con las otras subrutinas en alta resolución para el Commodore 64, el código máquina puede ser introducido también por medio de sentencias DATA si usted carece de un ensamblador. El listado inferior ha de entrarse por teclado y ejecutarse para poder cargar (LOAD) Circsub en memoria. Pero note que los cargadores en BASIC que afectan a Linesub y Circsub deben ser igualmente cargados y ejecutados antes que el programa de muestra. Una vez las tres rutinas en memoria, hay que eliminar las líneas 10, 15 y 20. También verá que el cargador en BASIC de Circsub ya incluye los *data* de la tabla de consulta, por lo cual no es necesario ejecutar en este caso el programa "creación tabla".

Muy pronto

Esta sección de lenguaje máquina ha analizado hasta el momento aquellos ordenadores provistos de un microprocesador Z80 o bien 6502. Nos queda un tercer microprocesador, igualmente popular, el chip 6809, que emplean el Dragon y el Tandy Color. Pronto nuestro curso centrará su atención en este chip para enseñar a programar en lenguaje máquina también a los usuarios del 6809



Programa de carga en BASIC

```
10 REM**** CARGADOR EN BASIC DE CIRCSUB ****
20 FORI=50432TO50432+498
30 READA:POKEI,A
40 CC=CC+A
50 NEXT I
60 READA:IFA<>CCTHENPRINT"ERROR EN SUMA DE CONTROL"
100 DATA0,63,89,108,123,137,149,159
110 DATA169,177,185,193,199,205,211
120 DATA216,221,226,230,233,237,240
130 DATA243,245,247,249,251,252,253
140 DATA254,255,255,255,255,255,254
150 DATA253,252,251,249,247,245,243
160 DATA240,237,233,230,226,221,216
170 DATA211,205,199,193,185,177,169
180 DATA159,149,137,123,108,89,63,0
190 DATA205,0,100,80,2,16,16,0,0,0,29
200 DATA1,31,1,100,100,100,100,0,0,120
210 DATA100,0,0,72,138,72,152,72,173
220 DATA68,197,41,31,141,70,197,173,68
230 DATA197,160,5,74,136,208,252,141
240 DATA69,197,173,65,197,56,237,68
250 DATA197,141,77,197,141,75,197,173
260 DATA66,197,233,0,141,78,197,141,76
270 DATA197,173,67,197,141,79,197,141
280 DATA80,197,169,0,170,141,71,197
290 DATA189,0,197,160,3,74,136,208,252
300 DATA141,87,197,189,0,197,41,7,141
310 DATA88,197,172,68,197,169,0,141,72
320 DATA197,141,73,197,141,74,197,173
330 DATA72,197,24,109,88,197,141,72
340 DATA197,201,8,144,23,56,233,8,141
350 DATA72,197,173,73,197,24,105,1,141
360 DATA73,197,173,74,197,105,0,141,74
370 DATA197,173,73,197,24,109,87,197
380 DATA141,73,197,173,74,197,105,0
390 DATA141,74,197,136,208,198,160,5
400 DATA78,74,197,110,73,197,110,72
410 DATA197,136,208,244,173,72,197,201
420 DATA16,144,9,173,73,197,24,105,1
430 DATA141,73,197,173,67,197,56,237
440 DATA73,197,141,81,197,173,81,197
450 DATA141,86,197,173,79,197,141,85
460 DATA197,32,165,198,173,67,197,24
470 DATA109,73,197,141,82,197,173,82
480 DATA197,141,86,197,173,80,197,141
490 DATA85,197,32,165,198,173,77,197
500 DATA141,75,197,173,78,197,141,76
510 DATA197,173,81,197,141,79,197,173
520 DATA82,197,141,80,197,173,71,197
530 DATA24,109,70,197,141,71,197,201
540 DATA32,144,26,173,71,197,56,233,32
550 DATA141,71,197,173,77,197,24,105,1
560 DATA141,77,197,173,76,197,105,0
570 DATA141,78,197,173,77,197,24,109
580 DATA69,197,141,77,197,173,78,197
590 DATA105,0,141,78,197,232,224,65
600 DATA240,3,76,153,197,104,168,104
610 DATA170,104,96,169,0,141,83,197
620 DATA141,84,197,173,75,197,205,77
630 DATA197,208,3,238,83,197,173,85
640 DATA197,205,86,197,208,3,238,84
650 DATA197,173,83,197,45,84,197,208
660 DATA39,173,75,197,141,0,195,173,76
670 DATA197,141,1,195,173,85,197,141,4
680 DATA195,173,77,197,141,2,195,173
690 DATA78,197,141,3,195,173,86,197
700 DATA141,5,195,32,14,195,96
710 DATA67223:REM"ERROR EN SUMA DE CONTROL"
```

Programa Circsub

```
                ;+++++++++++++
                ;+++++++++++++
                ;++            ==
                ;++  CIRCSUB 64  ++
                ;++            ++
                ;+++++++++++++
                ;+++++++++++++
                ;++++ VARIABLES LINESUB ++++

                LINSUB  = $C30E
                X1LO    = $C300
                X1HI    = $C301
                X2LO    = $C302
                X2HI    = $C303
                Y1      = $C304
                Y2      = $C305
                *       = $C500

                ;++++ VARIABLES CIRCSUB ++++

                TABLE   *=*+65          ;TABLA DE CONSULTA
                XCTRLO  *=*+1
                XCTRHI  *=*+1
                YCTR    *=*+1
                RADIUS  *=*+1
                INTR    *=*+1
                REMR    *=*+1
                TOTX    *=*+1
                RESREM  *=*+1
                RESLO   *=*+1
                RESHI   *=*+1
                OLDXLO  *=*+1
                OLDXHI  *=*+1
                NEWXLO  *=*+1
                NEWXHI  *=*+1
                OLDYA   *=*+1
                OLDYB   *=*+1
                NEWYA   *=*+1
                NEWYB   *=*+1
                XFLAG   *=*+1
                YFLAG   *=*+1
                OLDY    *=*+1
                NEWY    *=*+1
                INTTAB  *=*+1
                REMTAB  *=*+1

                ;++++ LLEVA REGISTROS A LA PILA ++++

                        PHA
                        TXA
                        PHA
                        TYA
                        PHA

                ;++++ CALC INT(R/32) Y RESTO ++++

AD 44 C5                LDA  RADIUS
29 1F                   AND  #$1F
8D 46 C5                STA  REMR
AD 44 C5                LDA  RADIUS
A0 05                   LDY  #$05
4A              LOOP1   LSR  A
88                      DEY
D0 FC                   BNE  LOOP1
8D 45 C5                STA  INTR

                ;++++ CALC. ABSCISA X INICIAL ++++

AD 41 C5                LDA  XCTRLO
38                      SEC
ED 44 C5                SBC  RADIUS
8D 4D C5                STA  NEWXLO
8D 4B C5                STA  OLDXLO
AD 42 C5                LDA  XCTRHI
E9 00                   SBC  #$00
8D 4E C5                STA  NEWXHI
8D 4C C5                STA  OLDXHI

                ;++++ ESTABLECE VALORES INICIALES OLD Y ++++

AD 43 C5                LDA  YCTR
8D 4F C5                STA  OLDYA
8D 50 C5                STA  OLDYB

                ;++++ CONTADOR A ZERO Y TOTX ++++

A9 00                   LDA  #$00
AA                      TAX
8D 47 C5                STA  TOTX

                ;++++ CALC. INTTAB Y REMTAB ++++

                NEXTPT
BD 00 C5                LDA  TABLE,X
A0 03                   LDY  #$03
                LOOP2
4A                      LSR  A
88                      DEY
D0 FC                   BNE  LOOP2
8D 57 C5                STA  INTTAB

BD 00 C5                LDA  TABLE,X
29 07                   AND  #$07
8D 58 C5                STA  REMTAB

                ;++++ MULTIPL. TABLA POR RADIO ++++

AC 44 C5                LDY  RADIUS
A9 00                   LDA  #$00
8D 48 C5                STA  RESREM
8D 49 C5                STA  RESLO          ;RESPUESTA 0
8D 4A C5                STA  RESHI

AD 48 C5        AGAIN   LDA  RESREM
18                      CLC
6D 58 C5                ADC  REMTAB
8D 48 C5                STA  RESREM
C9 08                   CMP  #$08           ;>=87
90 17                   BCC  NOCARR
38                      SEC
E9 08                   SBC  #$08           ;REINIC. REM
8D 48 C5                STA  RESREM
AD 49 C5                LDA  RESLO
18                      CLC
69 01                   ADC  #$01           ;INC RESULT.
8D 49 C5                STA  RESLO
AD 4A C5                LDA  RESHI
69 00                   ADC  #$00
8D 4A C5                STA  RESHI
                NOCARR
AD 49 C5                LDA  RESLO
18                      CLC
6D 57 C5                ADC  INTTAB
8D 49 C5                STA  RESLO
AD 4A C5                LDA  RESHI
69 00                   ADC  #$00
8D 4A C5                STA  RESHI
88                      DEY
D0 C6                   BNE  AGAIN

                ;++++ DIVIDE RESULTADO POR 32 ++++

A0 05                   LDY  #$05
                LOOP3
4E 4A C5                LSR  RESHI
6E 49 C5                ROR  RESLO
6E 48 C5                ROR  RESREM
88                      DEY
D0 F4                   BNE  LOOP3

AD 48 C5                LDA  RESREM
C9 10                   CMP  #$10           ; REM>=16?
90 09                   BCC  NOCRRY
AD 49 C5                LDA  RESLO
18                      CLC
69 01                   ADC  #$01           ; SUMAR 1
8D 49 C5                STA  RESLO

                ;++++ CALC. PRIMERA ORDENADA Y ++++

                NOCRRY
AD 43 C5                LDA  YCTR
38                      SEC
ED 49 C5                SBC  RESLO
8D 51 C5                STA  NEWYA

                ;++++ DIBUJA LÍNEA ++++

AD 51 C5                LDA  NEWYA
8D 56 C5                STA  NEWY
AD 4F C5                LDA  OLDYA
8D 55 C5                STA  OLDY
20 A5 C6                JSR  DRAW

                ;++++ CALC. 2.ª ORDENADA Y ++++

AD 43 C5                LDA  YCTR
18                      CLC
6D 49 C5                ADC  RESLO
8D 52 C5                STA  NEWYB

                ;++++ DIBUJA LÍNEA ++++

AD 52 C5                LDA  NEWYB
8D 56 C5                STA  NEWY
AD 50 C5                LDA  OLDYB
8D 55 C5                STA  OLDY
20 A5 C6                JSR  DRAW

                ;++++ CAMBIA OLD POR NEW ++++

AD 4D C5                LDA  NEWXLO
8D 4B C5                STA  OLDXLO
AD 4E C5                LDA  NEWXHI
8D 4C C5                STA  OLDXHI
AD 51 C5                LDA  NEWYA
8D 4F C5                STA  OLDYA
AD 52 C5                LDA  NEWYB
8D 50 C5                STA  OLDYB

                ;++++ ALTERA ABSCISA X ++++

AD 47 C5                LDA  TOTX
18                      CLC
6D 46 C5                ADC  REMR
8D 47 C5                STA  TOTX           ;SUM. RESTO A TOTAL
C9 20                   CMP  #$20           ;>32?
90 1A                   BCC  NOINC
AD 47 C5                LDA  TOTX
38                      SEC
E9 20                   SBC  #$20           ;REINIC. TOTX
8D 47 C5                STA  TOTX
AD 4D C5                LDA  NEWXLO
18                      CLC
69 01                   ADC  #$01           ;INC COORD X
8D 4D C5                STA  NEWXLO
AD 40 C5                LDA  OLDXHI
69 00                   ADC  #$00
8D 4E C5                STA  NEWXHI
                NOINC
AD 4D C5                LDA  NEWXLO
18                      CLC
6D 45 C5                ADC  INTR
8D 4D C5                STA  NEWXLO
AD 4E C5                LDA  NEWXHI
69 00                   ADC  #$00
8D 4E C5                STA  NEWXHI

                ;++++ INCREMENT COUNTER ++++

E8                      INX
E0 41                   CPX  #$41
F0 03                   BEQ  FINISH
4C 99 C5                JMP  NEXTPT

                ;++++ SACA REGISTROS DE LA PILA ++++

                FINISH
68                      PLA
A8                      TAY
68                      PLA
AA                      TAX
68                      PLA
60                      RTS

                ;++++ BUSCA EL MISMO PUNTO ++++

                DRAW
A9 00                   LDA  #$00
8D 53 C5                STA  XFLAG          ;FLAGS A 0
8D 54 C5                STA  XFLAG

AD 4B C5                LDA  OLDXLO
CD 4D C5                CMP  NEWXLO
D0 03                   BNE  NOXFLG
                NOXFLG
AD 55 C5                LDA  OLDY
CD 56 C5                CMP  NEWY
D0 03                   BNE  NOYFLG
EE 54 C5                INC  YFLAG
                NOYFLG
AD 53 C5                LDA  XFLAG
2D 54 C5                AND  YFLAG
D0 27                   BNE  NODRAW

                ;++++ DIBUJA LÍNEA ++++

AD 4B C5                LDA  OLDXLO
8D 00 C3                STA  X1LO
AD 4C C5                LDA  OLDXHI
8D 01 C3                STA  X1HI
AD 55 C5                LDA  OLDY
8D 04 C3                STA  Y1
AD 4D C5                LDA  NEWXLO
8D 02 C3                STA  X2LO
AD 4E C5                LDA  NEWXHI
8D 03 C3                STA  X2HI
AD 56 C5                LDA  NEWY
8D 05 C3                STA  Y2

20 0E C3                JSR  LINSUB
                NODRAW
60                      RTS
```

El sonido del éxito

Audiogenic se ha creado un prestigio como una de las principales firmas proveedoras de software de las máquinas Commodore



Nuevas oficinas
Las oficinas centrales de la empresa están situadas en Sutton Park, en las afueras de Reading. Audiogenic se trasladó allí en abril de 1984, abandonando una finca ubicada en el centro de Reading que ya no disponía de suficiente espacio de almacenamiento

Brillante gestión
El director gerente de Audiogenic, Martin Maynard, que fundó la empresa a principios de la década de los setenta como un estudio de grabación

El director gerente y fundador de Audiogenic, Martin Maynard, define la empresa a su cargo como un "negocio de comercialización y manufactura". Con anterioridad Maynard trabajaba en la industria de la música, y estableció Audiogenic en Reading (Gran Bretaña) a principios de la década de los setenta como un estudio de grabación y reproducción de cintas de audio. En 1978, Southern Electricity Board encargó a Audiogenic la duplicación de cassettes de datos para ordenador. El equipo con el que contaba la empresa se modificó para hacer frente a las demandas que implica la producción de cintas para ordenador a granel, y Audiogenic firmó un contrato con Commodore para ocuparse de la duplicación del software para el microordenador PET.

La empresa asumió entonces la comercialización y distribución del catálogo de cassettes de Commodore y empezó a vender libros, revistas y otros equipos relacionados con Commodore. Después del lanzamiento del Vic-20, en 1981, Maynard obtuvo licencias para comercializar los productos Vic desarrollados por firmas de software norteamericanas. El software fabricado bajo licencia supone todavía el 80 % del catálogo de Audiogenic y representa el 85-90 % del movimiento total de la empresa.

Este énfasis en la comercialización de software, en vez de la producción casera, parece haber tenido mucho éxito y Maynard calcula que hasta ahora Audiogenic ha producido más de un millón de cassettes. "Nuestro punto más fuerte es que tenemos un catálogo enorme. Esta diversidad significa que podemos comprender lo que va sucediendo en el mercado", explica Maynard. "Habiendo estado en la industria del software durante seis años, ya lo hemos visto todo. De todo el software que se distribuyó el año pasado, entre el 20 y el 30 % todavía está en las estanterías de alguien. Los escritores están perdiendo el contacto con lo que realmente desea la gente."



Este cauteloso enfoque, promocionando software ya probado, contrasta firmemente con las tácticas improvisadas que aplican muchas otras firmas de software.

En la actualidad Audiogenic tiene una plantilla de 25 personas. El principal programador de la empresa, Dave Middleton (escritor del paquete de base de datos Magpie, tan bien conceptuado) trabaja en calidad de colaborador independiente. Audiogenic tiende a concentrarse en el software de utilidades para sus propios programas, en vez de perseguir los rápidos beneficios que se suelen obtener en el mercado de juegos. Como explica Maynard, "los ordenadores siempre tendrán un elemento de juegos, pero esa fase ahora está muriendo y el software para ordenador se irá desarrollando hasta convertirse en algo más útil. Cuando estás vendiendo doscientos o trescientos ejemplares de un paquete por mes, piensas que no se está vendiendo bien, pero al cabo de un año todavía se estarán vendiendo en esa misma cantidad".

Esto no significa que la empresa rechace de plano el mercado de juegos. Uno de los mayores éxitos de ventas de Audiogenic fue *Motor mania* y recientemente la empresa ha lanzado *Alice in Videoland* (Alicia en el País del Video) para el Commodore 64. Desarrollado bajo licencia, este juego consta de un impresionante programa de 90 Kbytes distribuido en cinco pantallas que se cargan desde disco a medida que va avanzando el juego.

Al poco tiempo de haberse trasladado a unas instalaciones más grandes, en marzo de 1984, Audiogenic instaló equipos de reproducción de cintas modernizados. La nueva maquinaria reproduce un programa repetidamente en una cinta continua, que se corta y se empaqueta en forma de cassette después de la duplicación. Este sistema es más rápido y también más conveniente que el sistema antiguo, que reproducía los programas en cassettes individuales e implicaba que la empresa necesitaba tener almacenadas grandes existencias de cassettes C10 y C20 vírgenes.

Audiogenic pretende continuar su política de distribuir productos fabricados por otras empresas en Gran Bretaña y Estados Unidos. La diversificación de hardware de periféricos ha llevado a la empresa a comercializar una pastilla para gráficos que desarrolló Koala Technologies. La planificación de proyectos de software incluye una gama de cassettes para máquinas MSX y software para el nuevo ordenador personal Commodore 16.

Editorial Delta, S.A.



10047
9 788485 822836